cnit

18-19 Novembre 2008

Aula Comunicazioni (lato reception) - Edificio 2 – Parco Scientifico – PULA

Seminario sullo Standard MPEG-4

Utilizzo ed aspetti implementativi

a cura di:
CNIT (Consorzio Nazionale Interuniversitario per le Telecomunicazioni)
Relatore: Ing. Cristian Perra

18 Novembre 2008 10:00 - 13:30

- Il gruppo MPEG
- Elementi di lavoro
- MPEG 4 Overview
- Miglioramento dell'efficienza di codifica: caratteristiche innovative introdotte in MPEG4 AVC
- Algoritmo di codifica MPEG 4 AVC (parte 1): aspetti implementativi relativamente alla predizione e trasformazione

19 Novembre 2008 10:00 - 13:30

- Algoritmo di codifica MPEG 4 AVC (parte 2): aspetti implementativi relativamente alla quantizzazione, deblocking filtraggio, robustezza all'errore
- MPEG 4 Network Abstraction Layer
- MPEG 4 Elementary Stream
- MPEG 4 Transport Stream
- Sessione di prova e valutazione delle performance

Ing. Cristian Perra (cristian.perra@cnit.it)

cnit

Introduzione

Una delle tecnologie chiave che hanno permesso il grande sviluppo della televisione digitale è la compressione video. La tecnologia di codifica video nota come MPEG-2, sviluppata nei primi anni novanta, è diventata lo standard di trasmissione DTV (digital television) sia satellitare che terrestre in quasi tutti i paesi del mondo. Da allora la velocità dei microprocessori e le capacità di memoria dei dispositivi hardware per la codifica e la decodifica sono migliorate significativamente rendendo possibile lo sviluppo e l'implementazione di algoritmi di codifica innovativi in grado di abbattere significativamente i limiti di compressione dello standard MPEG-2.

cnit

Introduzione

Tali innovazioni, sfociate nel 2003 nello standard MPEG-4 Advanced Video Coding, non hanno permesso di mantenere la compatibilità all'indietro con l'MPEG-2, e questo ha inizialmente costituito un limite alla loro introduzione nei sistemi di trasmissione DTV.

Tuttavia, negli ultimi anni la codifica MPEG-4 Advanced Video Coding (AVC) si è diffusa rapidamente, è stata adottata da DVB, recentemente da ATSC, ed è lo standard di codifica nell'IPTV.

L'obiettivo di questo seminario è quello di presentare lo standard di codifica MPEG-4 AVC con particolare attenzione agli aspetti implementativi del livello di codifica video.

cnit

Il gruppo MPEG

- **Moving Picture Experts Group** (MPEG)

  E' un working group dell'ISO/IEC con l'incarico di sviluppare standard internazionali per:

  –Compressione

  –Decompressione

  –Elaborazione

  –Rappresentazione codificata

  di immagini in movimento, audio e loro combinazioni

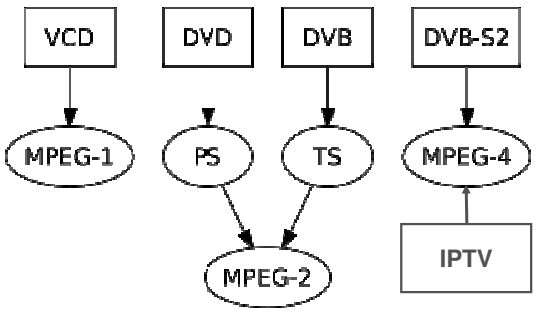
cnit

Meeting

- Il gruppo MPEG si incontra solitamente 3 volte all'anno:
  - Riunioni plenarie
  - Riunioni dei sottogruppi :
    - Requirements, Systems, Multimedia Description Schemes, Video, Audio, Synthetic Natural Hybrid Coding, Test, Implementation Studies and Liaison.
- Ai meeting MPEG partecipano oltre 300 esperti provenienti da più di 20 paesi.

cnit

MPEG e Video Digitale



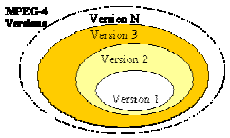
Prodotti MPEG

- **MPEG-1**, lo standard per l'archiviazione l'estrazione di immagini in movimento e audio su supporti di memorizzazione (approvato a Novembre 1992)
- **MPEG-2**, lo standard per la televisione digitale (approvato a Novembre 1994)

Prodotti MPEG

- **MPEG-4**, lo standard per applicazioni multimediali
  - Versione 1 approvata in Ottobre 1998
  - Versione 2 approvata a Dicembre 1999

▪Dopo queste due versioni principali sono stati introdotti nuovi 'tool' in emendamenti che possono essere qualificati come versioni.
▪Non è tanto importante riconoscere le versioni quanto distinguere i 'profile' (profili). I tool esistenti e i profili di ogni versione non sono mai sostituiti in versioni successive.
▪Le tecnologie sono sempre aggiunte in MPEG-4 sotto forma di nuovi profili. Questo permette la compatibilità all'indietro delle nuove versioni.



Prodotti MPEG

- **MPEG-7**, lo standard per la rappresentazione dei contenuti per ricerca, filtraggio, gestione e elaborazione di informazioni multimediali (approvato in Ottobre 2001)
- **MPEG-21**, il framework del multimedia

Prodotti MPEG

- MPEG-A: Multimedia application format.
- MPEG-B: MPEG systems technologies.
- MPEG-C: MPEG video technologies.
- MPEG-D: MPEG audio technologies.
- MPEG-E: Multimedia Middleware.

11172 (MPEG-1)

- Coding of moving pictures and associated audio at up to about 1.5 Mbit/s
  - Part 1 Systems
  - Part 2 Video
  - Part 3 Audio (Layer I, II, III (MP3))
  - Part 4 Conformance testing
  - Part 5 Software simulation

13818 (MPEG-2)

- Generic coding of moving pictures and associated audio:
  - Part 1 Systems
  - Part 2 Video
  - Part 3 Audio
  - Part 4 Conformance testing
  - Part 5 Software simulation

13818 (MPEG-2) (cont)

- Part 6 System extensions - DSM-CC
- Part 7 Advanced Audio Coding
- Part 8 VOID - (withdrawn)
- Part 9 System extension RTI
- Part 10 Conformance extension - DSM-CC
- Part 11 IPMP on MPEG-2 Systems

14496 (MPEG-4)

- **Coding of audio-visual objects**
  - Parte 1 – Systems
  - **Parte 2 – Visual**
  - Parte 3 – Audio
  - ….
  - **Parte 10 – Advanced Video Coding**
  - Parte 23 - Symbolic Music Representation

Parte 1 ➔ ISO/IEC 14496-1

- **Systems** / Sistema
- Descrive la sincronizzazione e il multiplexing dell'audio e del video. Per esempio il TS (transport stream).

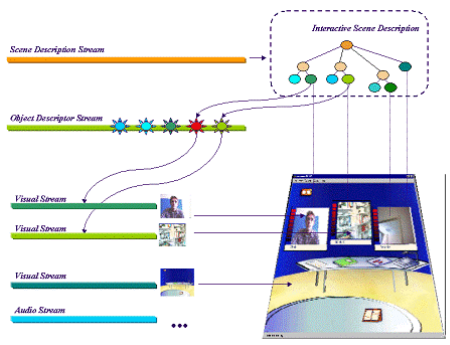
Parte 1 ➔ ISO/IEC 14496-1

- La codifica produce un flussi di dati elementari (dati video, dati audio, ecc.) detti ES (elementary stream)
- Gli ES possono essere trasmessi o archiviati
- Gli ES necessitano di un processo di composizione che permetta la costruzione della corretta presentazione da parte del ricevitore.
- MPEG-4 system descrive le relazione tra i componenti audio e video che costituiscono una scena attraverso BIFS e OD.

Parte 1 ➔ ISO/IEC 14496-1

- Relazione tra i componenti audio e video che costituiscono una scena.
- Due livelli di descrizione per tali relazioni:
- BIFS (Binary Format for Scenes)
  - Descrive le relazione spazio-temporali tra gli oggetti della scena
- ODs (Object Descriptors)
  - Definisce le relazioni tra gli ES relativi a ogni oggetto (ODs) define the relationship between the Elementary Streams pertinent to each object (e.g the audio and the video stream of a participant to a videoconference) ODs also provide additional information such as the URL needed to access the Elementary Steams, the characteristics of the decoders needed to parse them, intellectual property and others.

Parte 1 ➔ ISO/IEC 14496-1



Parte 2 ➔ ISO/IEC 14496-2

- **Visual (Advanced Simple Profile)**
- Un codec per la compressione di dati video, tessiture (texture), immagini sintetiche, ecc. Per esempio nella parte 2 è descritto il “profilo” ASP (Advanced Simple Profile) per la codifica video.

Parte 3 ➔ISO/IEC 14496-3

- **Audio**
- Un insieme di codec che sfuttano modelli di percezione audio per la compressione di segnali audio. Per esempio in questa parte sono descritte alcune varianti dell’AAC (Advanced Audio Coding) e altri strumenti di codifica per l’audio/parlato.

Parte 4 ➔ ISO/IEC 14496-4

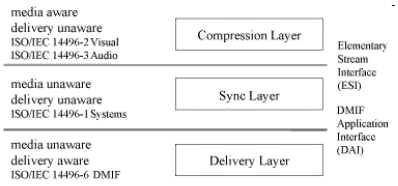
- **Conformance**
- Descrive le procedure per verificare la conformità rispetto alle altri parti dello standard.

Parte 5 ➔ ISO/IEC 14496-5

- **Reference Software**
- Fornisce il software dimostrativo per lo sviluppo delle parti dello standard.

Parte 6 ➔ ISO/IEC 14496-6

- DMIF (Delivery Multimedia Integration Framework)
  - Interfaccia tra applicazione e trasporto
  - Una singola applicazione può essere eseguita su diversi livelli di trasporto se supportata dalla corretta istanza del DMIF.



Parte 7 ➔ ISO/IEC 14496-7

- **Optimized Reference Software**
- Fornisce esempi su come migliorare l’implementazione (p.e., in relazione alla Parte 5)

Parte 8 ➔ ISO/IEC 14496-8

- **Carriage on IP networks**
- Specifica un metodo per il trasporto di contenuti MPEG-4 su reti IP

Parte 9 ➔ ISO/IEC 14496-9

- **Reference Hardware**
- Fornisce progetti hardware per dimostrare come implementare le altre parti dello standard.

Parte 10 ➔ ISO/IEC 14496-10

- Advanced Video Coding (AVC)
- Un codec per segnali video.
- Sviluppato congiuntamente con l'ITU e standardizzato da ITU come ITU-T H.264. (JVT, Joint Video Team)
- Quindi i seguenti standard descrivono la medesima tecnologia:
  - MPEG-4 Part 10 AVC, oppure MPEG4-AVC
  - H.264

Parte 11 ➔ISO/IEC 14496-11

- Scene description and Application engine ("*BIFS*")
- Sviluppata per realizzare contenuti interattivi con più profili e versioni sia 2D che 3D.

Parte 12 ➔ ISO/IEC 14496-12

- ISO Base Media File Format
- Un formato di file per l'archiviaizone di contenuti multimediali.

Parte 13 ➔ ISO/IEC 14496-13

- Intellectual Property Management and Protection (IPMP) Extensions.

Parte 14 ➔ ISO/IEC 14496-14

- MPEG-4 File Format
- Formato di file contenitore per contenuti MPEG-4, basato sulla parte 12.

Parte 15 ➔ ISO/IEC 14496-15

- AVC File Format
- Per l'archiviazione del video della Parte 10, basato sulla Parte 12.

Parte 16 – Parte 19

- Parte 16 ➔ ISO/IEC 14496-16
  - Animation Framework eXtension (AFX)
- Parte 17 ➔ ISO/IEC 14496-17
  - Timed Text subtitle format.
- Parte 18 ➔ ISO/IEC 14496-18
  - Font Compression and Streaming (for OpenType fonts).
- Parte 19 ➔ ISO/IEC 14496-19
  - Synthesized Texture Stream.

Parte 20 – Parte 23

- Parte 20 ➔ ISO/IEC 14496-20
  - Lightweight Application Scene Representation (LASeR).
- Parte 21 ➔ ISO/IEC 14496-21
  - MPEG-J Graphical Framework eXtension (GFX)(not yet finished - at "FCD" stage in July 2005, FDIS January 2006).
- Parte 22 ➔ ISO/IEC 14496-22
  - Open Font Format Specification (OFFS) based on OpenType(not yet finished - reached "CD" stage in July 2005)
- Parte 23 ➔ ISO/IEC 14496-23
  - Symbolic Music Representation (SMR)(not yet finished - reached "FCD" stage in October 2006)

MPEG-4 Trasporto

- MPEG-4 non definisce livelli di trasporto
- In alcuni casi è stato definito un adattamento a specifici livelli di trasporto:
  - Trasporto su MPEG-2 Transport Stream (come emendamento al MPEG-2 Systems)
  - Trasporto su IP (in cooperazione con IETF)

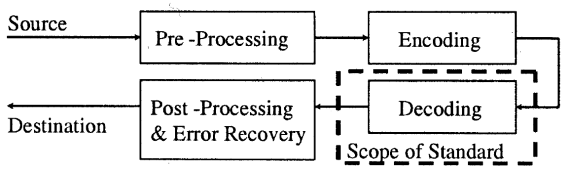
MPEG-4 Overview

- Gruppi di studio internazionale hanno lavorato su tecniche di codifica video dal 1990:
  - VCEG (Video Coding Experts Group) dell'ITU-T (International Telecommunication Union - Telecommunication sector)
  - MPEG (Moving Picture Experts Group) dell'ISO/IEC
- ITU-T ha sviluppato H.261 come primo standard di codifica video per applicazioni di videoconferenza
- MPEG-1 (standard di codifica video video) è stato realizzato per l'archiviazione su CD
- MPEG-2 (adottato da ITU-T come H.262)
  - Estensione di MPEG-1
  - Standard per TV digitale e HDTV as extension of MPEG-1

Introduzione

- MPEG-4 parte 2
  - Codifica classica
  - Codifica a oggetti
  - Codifica di video/audio naturale e sintetico
- ITU-T ha sviluppato H.263 per migliorare le prestazioni della compressione video rispetto a H.261;
- Il modello di codifica H.263 è stato adottato come core per l'MPEG-4 parte 2
- Nota: MPEG 1,2 e 4 standardizzano anche la codifica audio.

Standardizzazione

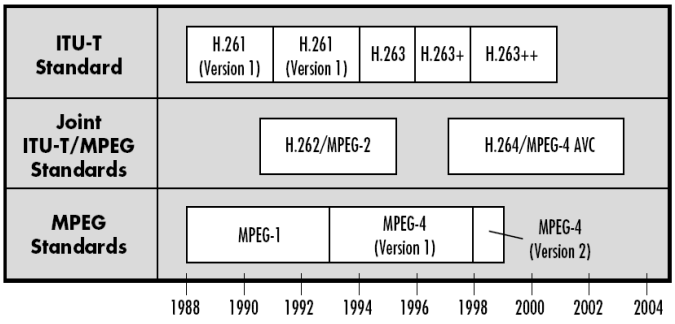


- Definizione del bitstream
- Definizione della sintassi
- Definizione del processo di decodifica degli elementi della sintassi
- Massima libertà nell'ottimizzazione delle implementazioni per applicazioni specfiche
- Nessuna garanzia della qualità di riproduzione end-to-end

Introduzione

- H.264 / MPEG-4 parte 10 (Advanced Video Coding, AVC)
  - Standard per la codifica video
  - Sviluppato dal JVT (Joint Video Team) composto da esperti del VCEG and MPEG.
  - Obiettivo: migliorare ulteriormente la capacità di compressione video rispetto agli standard esistenti:
    - Codifica estremamente efficiente
    - Specifiche di sintassi semplici
    - Pensato per integrazione della codifica con i protocolli e le architetture di multiplexing esistenti
- H.264 supporta:
  - video broadcasting – video streaming - video conferencing
  - Su reti fisse e reti wireless mediante diversi protocolli di trasporto

Storia

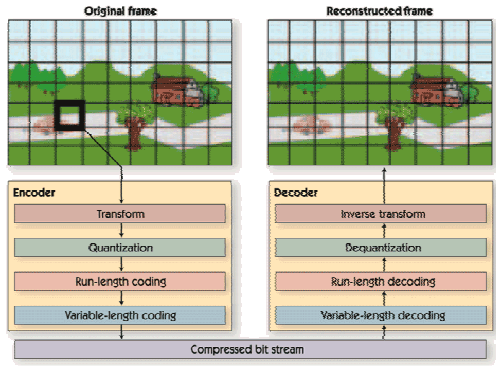


Elementi di base funzionali

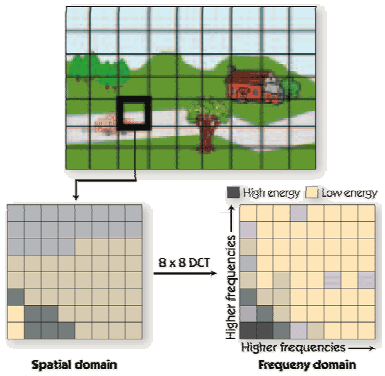
- Trasformata per riduzione della correlazione spaziale
- Quantizzatore per controllo del rate (bitrate)
- Compensazione del moto per riduzione della correlazione temporale
- Codifica entropica per riduzione della correlazione statistica
- Nota: H.264 ha gli stessi elementi di base funzionale degli standard precedenti: MPEG-1, MPEG-2, MPEG-4 part 2, H.261, H.263.

Sequenza video

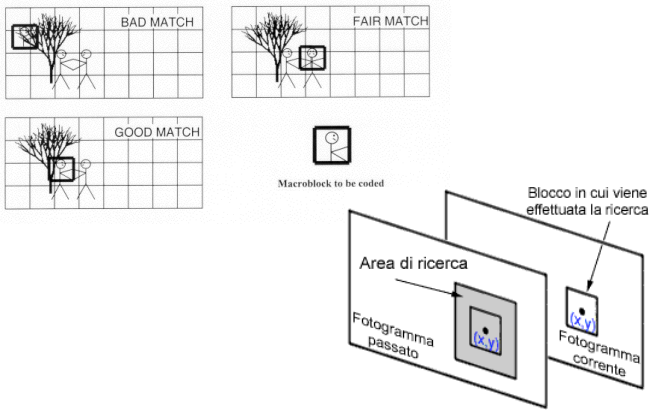
Intra coding



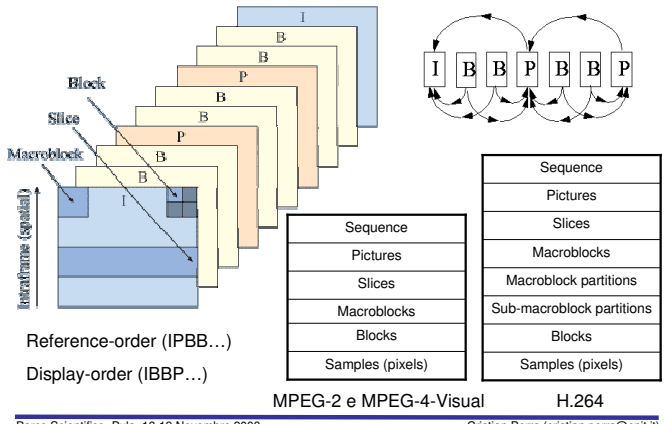
Trasformata



Compensazione del moto



Frame I,P,B e gerarchia



Reference-order (IPBB…)

Display-order (IBBP…)

MPEG-2 e MPEG-4-Visual H.264

Cambiamenti importanti

- Dimensione dei blocchi per la compensazione del moto variabile
- Precisione della compensazione del moto al quarto di pixel
- Vettori di moto oltre i bordi immagine
- Più frame di riferimento per la compensazione del moto
- Disaccoppiamento del reference-order dal display-order
- Disaccoppiamento dai metodi di rappresentazione dell'immagine dalla possibilità di essere usata come riferimento per la compensazione del moto
- Predizione pesata
- Miglioramento del metodo di inferenza del moto nelle aree non codificate

Efficienza di codifica

- L'efficienza di codifica è aumentata a spese dell'aumento di complessità del encoder/decoder. H.264 sfrutta alcuni metodi per ridurre la complessità computazionale:
  - Trasformata intera priva di moltiplicazioni
  - Operazione di prodotto nella trasformata combinata con operazioni di prodotto nel quantizzatore

Robustezza e flessibilità

- Parameter Set Structure
- Strutture di sintassi incapsulate da unità NAL
- Dimensione flessibile delle slice
- Flexible Macroblock Ordering (FMO)
- Arbitrary Slice Ordering (ASO)
- Redundant Pictures
- Data Partitioning
- SP/SI syncronization/switching pictures

Efficienza di codifica

- Canali rumorosi (p.e. reti wireless) impediscono al decoder una corretta ricezione del bitstream video codificato. La decodifica errata degrada la qualità soggettiva dell'immagine e si propaga nei blocchi o picture seguenti.
  - In aggiunta al data partitioning (p.e. usato in MPEG-4 parte 2) H.264 si serve di metodi per l'error-resilience:
    - parameter setting (PS)
    - flexible macroblock ordering (FMO)
    - switched slice (SS)
    - redundant slice(RS)

Profili e Livelli

- Come in altri standard MPEG anche H.264 definisce Profili e Livelli che specificano restrizioni sul bitstream
- Sette profili coprono applicazioni da reti wireless a cinema digitale
- Nota: altre tecniche di codifica video che usano gli stessi blocchi funzionali dell'H.264 con minime modifiche sono:
  - Microsoft Windows Media Video 9 (WMV-9) della Society of Motion Picture and Television Engineers (SMPTE)
  - AVS (Audio Video Coding Standard) dalla Cina.

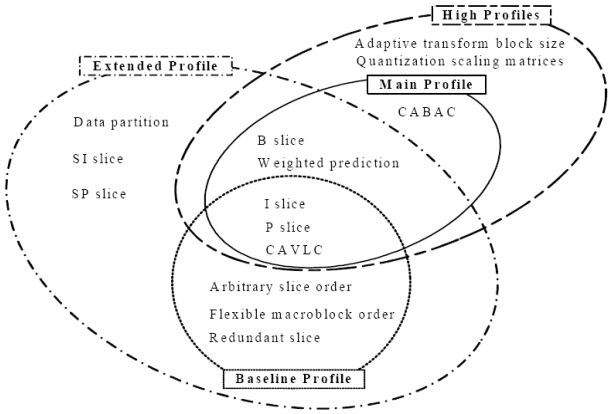
Profili

- Ogni profilo specifica la sintassi di un sottoinsieme dell'intero bitstream e specifica le funzionalità che dovrebbero essere supportate da tutti i decoder conformi a quel profilo.
- **H.264 prima versione (tre profili):**
  - **Baseline**, servizi conversazionali in tempo reali (video conferenza, video telefonia)
  - **Main**, archiviazione e broadcast televisivo
  - **Extended**, servizi multimediali su Internet

Profili

- **High Profiles - Fidelity range extensions:**
  - **High Profiles** defined in the **fidelity range extensions** for applications such as content-contribution, content-distribution and studio editing and post-processing
    - **High Profile** is to support the 8-bit video with 4:2:0 sampling for applications using high resolution.
    - **High 10 Profile** is to support the 4:2:0 sampling with up to 10 bits of representation accuracy per sample.
    - **High 4:2:2 Profile** is to support up to 4:2:2 chroma sampling and up to 10 bits per sample.
    - **High 4:4:4 Profile** is to support up to 4:4:4 chroma sampling, up to 12 bits per sample, and integer residual color transform for coding RGB signal.
- The Profiles have both the common coding parts.

Profili e Livelli



Coding Tools

Coding Tools	Baseline	Main	Extended
Slice I e P	X	X	X
CAVLC	X	X	X
CABAC		X	
Slice B		X	X
Interlaced Coding (PicAFF, MBAFF)	X	X	X
Enhanced Error Resilience (FMO, ASO, RS)			X
Further Enhanced Error Resilience (DP)			X
Slice SP e SI			X

Parti comuni a tutti i profili

- Slice I (Intra-coded slice), la codifica sfrutta la predizione da campioni appartenenti alla stessa slice
- Slice P (Predictive-coded slice), la codifica sfrutta la predizione da blocchi appartenenti a picture di riferimento già codificate (un motion vector e reference index)
- CAVLC (Context-based Adaptive Variable Length Coding) per la codifica entropica

SLICE
Each **picture** is compressed by partitioning it as one or more **slices**; each slice consists of macroblocks, which are blocks of 16x16 luma samples with corresponding chroma samples.

Baseline Profile

- Flexible macroblock order: macroblocks may not necessarily be in the raster scan order. The map assigns macroblocks to a slice group.
- Arbitrary slice order: the macroblock address of the first macroblock of a slice of a picture may be smaller than the macroblock address of the first macroblock of some other preceding slice of the same coded picture.
- Redundant slice: This slice belongs to the redundant coded data obtained by same or different coding rate, in comparison with previous coded data of same slice.

Main Profile

- B slice (Bi-directionally predictive-coded slice): the coded slice by using inter prediction from previously-decoded reference pictures, using at most two motion vectors and reference indices to predict the sample values of each block.
- Weighted prediction: scaling operation by applying a weighting factor to the samples of motion compensated prediction data in P or B slice.
- CABAC (Context-based Adaptive Binary Arithmetic Coding) for entropy coding

Extended Profile

- Includes all parts of Baseline Profile:
  - flexible macroblock order
  - arbitrary slice order
  - redundant slice
- SP slice: the specially coded slice for efficient switching between video streams, similar to coding of a P slice.
- SI slice: the switched slice, similar to coding of an I slice.
- Data partition: the coded data is placed in separate data partitions, each partition can be placed in different layer unit.
- B slice
- Weighted prediction

High Profiles

- Includes all parts of Main Profile : B slice, weighted prediction, CABAC
- Adaptive transform block size : 4 x 4 or 8 x 8 integer transform for luma samples
- Quantization scaling matrices : different scaling according to specific frequency associated with the transform coefficients in the quantization process to optimize the subjective quality

Coding Tools High Profiles

Coding Tools	High	High 10	High 4:2:2	High 4:4:4
Main Profile Tools	X	X	X	X
4:2:0 Chroma Format	X	X	X	X
8 Bit Sample Bit Depth	X	X	X	X
8x8 vs. 4x4 Transform Adaptivity	X	X	X	X
Quantization Scaling Matrices	X	X	X	X
Separate Cb and Cr QP control	X	X	X	X
Monochrome video format	X	X	X	X
9 and 10 Bit Sample Bit Depth		X	X	X
4:2:2 Chroma Format			X	X
11 and 12 Bit Sample Bit Depth				X
4:4:4 Chroma Format				X
Residual Color Transform				X
Predictive Lossless Coding				X

Application requirements

Application	Requirements	H.264 Profiles (MPEG-4 Part 10)	MPEG-4 Part 2 Profiles
Broadcast television	Coding efficiency, reliability (over a controlled distribution channel), interlace, low-complexity decoder	MAIN	ASP
Streaming video	Coding efficiency, reliability (over a uncontrolled packet-based network channel), scalability	EXTENDED	FGS
Video storage and playback	Coding efficiency, interlace, low-complexity encoder and decoder	MAIN	ASP
Videoconferencing	Coding efficiency, reliability, low latency, low-complexity encoder and decoder	BASELINE	SP
Mobile video	Coding efficiency, reliability, low latency, low-complexity encoder and decoder, low power consumption	BASELINE	SP
Studio distribution	Lossless or near-lossless, interlace, efficient transcoding	MAIN HIGH PROFILE	STUDIO

(SP, ASP, ARTS, FGS, Studio : Simple, Advanced Simple, Advanced Real Time Simple, Fine Granular Scalability, and Studio Profiles)

Levels

- For any given Profile, Levels generally correspond to processing power and memory capability of a codec.
- Each Level may support a different picture size – QCIF, CIF, ITU-R 601 (SDTV), HDTV, S-HDTV, DCinema
- Also each Level sets the limits for data bitrate, frame size, picture buffer size, etc.

Levels

- In the standard, *Levels* specify the **maximum frame size in terms of only the total number of pixels/frame**.
- Horizontal and Vertical maximum sizes are not specified except for constraints that horizontal and vertical sizes can not be more than Sqrt(maximum frame size * 8).
- If, at a particular level, the picture size is less than the one in the table, then a correspondingly larger number of reference pictures (up to 16 frames) can be used for motion estimation and compensation.
- Similarly, instead of specifying a maximum frame rate at each level, a maximum sample (pixel) rate, in terms of macroblocks per second, is specified.
- Thus if the picture size is smaller than the typical pictures size in the following table, then the frame rate can be higher than that in the table, all the way up to a maximum of 172 frames/sec.

Levels

Level number	Max macroblocks per second	Max frame size (macroblocks)	Max video bit rate (VCL) for Baseline, Extended and Main Profiles	Max video bit rate (VCL) for High Profile	Max video bit rate (VCL) for High 10 Profile	Max video bit rate (VCL) for High 4:2:2 and High 4:4:4 Predictive Profiles	Examples for high resolution @ frame rate (max stored frames) in Level
1	1485	99	64 kbit/s	80 kbit/s	192 kbit/s	256 kbit/s	128x96@30.9 (8) 176x144@15.0 (4)
1b	1485	99	128 kbit/s	160 kbit/s	384 kbit/s	512 kbit/s	128x96@30.9 (8) 176x144@15.0 (4)
1.1	3000	396	192 kbit/s	240 kbit/s	576 kbit/s	768 kbit/s	176x144@30.3 (9) 320x240@10.0 (3) 352x288@7.5 (2)
1.2	6000	396	384 kbit/s	480 kbit/s	1152 kbit/s	1536 kbit/s	320x240@20.0 (7) 352x288@15.2 (6)
1.3	11880	396	768 kbit/s	960 kbit/s	2304 kbit/s	3072 kbit/s	320x240@36.0 (7) 352x288@30.0 (6)
2	11880	396	2 Mbit/s	2.5 Mbit/s	6 Mbit/s	8 Mbit/s	320x240@36.0 (7) 352x288@30.0 (6)
2.1	19800	792	4 Mbit/s	5 Mbit/s	12 Mbit/s	16 Mbit/s	352x480@30.0 (7) 352x576@25.0 (6)
2.2	20250	1620	4 Mbit/s	5 Mbit/s	12 Mbit/s	16 Mbit/s	352x480@30.7(10) 352x576@25.6 (7) 720x480@15.0 (6) 720x576@12.5 (5)

Levels

Level number	Max macroblocks per second	Max frame size (macroblocks)	Max video bit rate (VCL) for Baseline, Extended and Main Profiles	Max video bit rate (VCL) for High Profile	Max video bit rate (VCL) for High 10 Profile	Max video bit rate (VCL) for High 4:2:2 and High 4:4:4 Predictive Profiles	Examples for high resolution @ frame rate (max stored frames) in Level
3	40500	1620	10 Mbit/s	12.5 Mbit/s	30 Mbit/s	40 Mbit/s	352x480@61.4 (12) 352x576@51.1 (10) 720x480@30.0 (6) 720x576@25.0 (5)
3.1	108000	3600	14 Mbit/s	17.5 Mbit/s	42 Mbit/s	56 Mbit/s	720x480@80.0 (13) 720x576@66.7 (11) 1280x720@30.0 (5)
3.2	216000	5120	20 Mbit/s	25 Mbit/s	60 Mbit/s	80 Mbit/s	1280x720@60.0 (5) 1280x1024@42.2 (4)
4	245760	8192	20 Mbit/s	25 Mbit/s	60 Mbit/s	80 Mbit/s	1280x720@68.3 (9) 1920x1080@30.1 (4) 2048x1024@30.0 (4)
4.1	245760	8192	50 Mbit/s	50 Mbit/s	150 Mbit/s	200 Mbit/s	1280x720@68.3 (9) 1920x1080@30.1 (4) 2048x1024@30.0 (4)
4.2	522240	8704	50 Mbit/s	50 Mbit/s	150 Mbit/s	200 Mbit/s	1920x1080@64.0 (4) 2048x1080@60.0 (4)
5	589824	22080	135 Mbit/s	168.75 Mbit/s	405 Mbit/s	540 Mbit/s	1920x1080@72.3 (13) 2048x1024@72.0 (13) 2048x1080@67.8 (12) 2560x1920@30.7 (5) 3680x1536@26.7 (5)
5.1	983040	36864	240 Mbit/s	300 Mbit/s	720 Mbit/s	960 Mbit/s	1920x1080@120.5 (16) 4096x2048@30.0 (5) 4096x2304@26.7 (5

H.264/AVC Levels

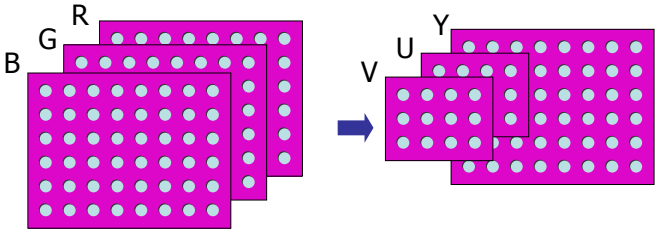
Level Number	Typical Picture Size	Typical frame rate	Maximum compressed bit rate (for VCL) in Non-FRExt profiles	Maximum number of reference frames for typical picture size
1	QCIF	15	64 kbps	4
1b	QCIF	15	128 kbps	4
1.1	CIF or QCIF	7.5 (CIF) / 30 (QCIF)	192 kbps	2 (CIF) / 9 (QCIF)
1.2	CIF	15	384 kbps	6
1.3	CIF	30	768 kbps	6
2	CIF	30	2 Mbps	6
2.1	HHR (480i or 576i)	30 / 25	4 Mbps	6
2.2	SD	15	4 Mbps	5
3	SD	30 / 25	10 Mbps	5
3.1	1280x720p	30	14 Mbps	5
3.2	1280x720p	60	20 Mbps	4
4	HD Formats (720p or 1080i)	60p / 30i	20 Mbps	4
4.1	HD Formats (720p or 1080i)	60p / 30i	50 Mbps	4
4.2	1920x1080p	60p	50 Mbps	4
5	2kx1k	72	135 Mbps	5
5.1	2kx1k or 4kx2k	120 / 30	240 Mbps	5

Hierarchy of a video sequence

Sequence
Pictures
Slices
Macroblocks
Macroblock partitions
Sub-macroblock partitions
Blocks
Samples (pixels)

Color Space

- Because human visual system is more sensitive to *luma* than *chroma,* in H264/AVC C_b, C_r components are subsampled and have ¼ of the number of samples than the Y component ➡ 4:2:0 sampling

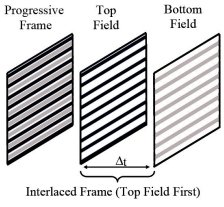


Video Coding Layer

- VCL contains the core video coded data, which consists of video sequence, picture, slice, and macroblock. The video sequence has either frames or fields which are comprised of three sample arrays, one luma and two chroma sample arrays or the RGB arrays (High 4:4:4 Profile only).
- The standard supports either progressive-scan or interlaced-scan, which may be mixed together in the same sequence.
- Baseline Profile is limited to progressive scan.

Pictures, frames, and fields

- A coded pictures can represent either an entire frame or a single field
- A frame of video can be considered to contain two interleaved fields
  - interlaced frame: the two fields of a frame were captured at different time instants
  - progressive frame
- The coding representation in H.264/AVC is primarily agnostic with respect to this video characteristic

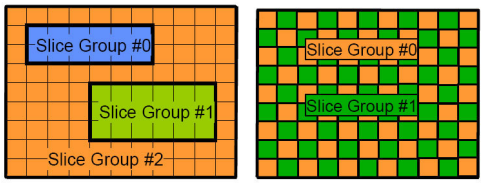


Adaptive frame/field coding operation

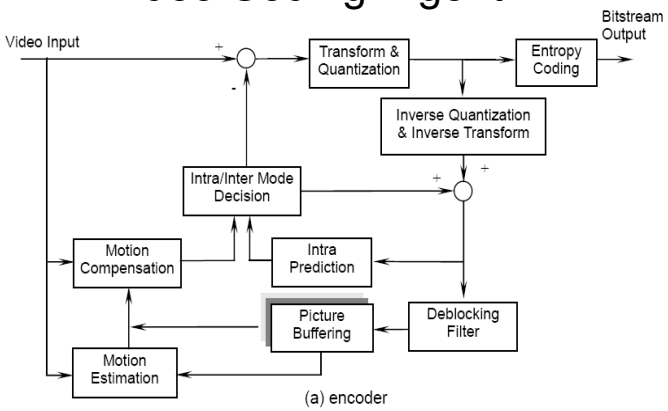
- In interlaced frames with regions of moving objects, two adjacent rows tend to show a reduced degree of statistical dependency
- H.264/AVC design allows any of the following decisions for coding a frame:
  - Frame mode: combine the two fields together
  - Field mode: not combine the two fields together
- The choice can be made adaptively for each frame and is referred to as Picture Adaptive Frame/Field (PAFF) coding
- Field mode:
  - Motion compensation utilizes reference fields
  - De-blocking filter is not used for horizontal edges of macroblocks
    - Moving region → field mode
    - Non-moving region → frame mode

Video Coding Layer

- Pictures are divided into slices. A slice is a sequence of macroblocks and has the flexible size, especially one slice within a picture. In case of multiple slice groups, the allocation of macroblocks is determined by a macroblock to slice group map that indicates which slice group each MB belongs to.
- In the 4:2:0 format, each macroblock is comprised of one 16 x 16 luma and two 8 x 8 chroma sample arrays. In the 4:2:2 format, the chroma sample arrays are 8 x 16, and in the 4:4:4, the arrays are 16 x 16.

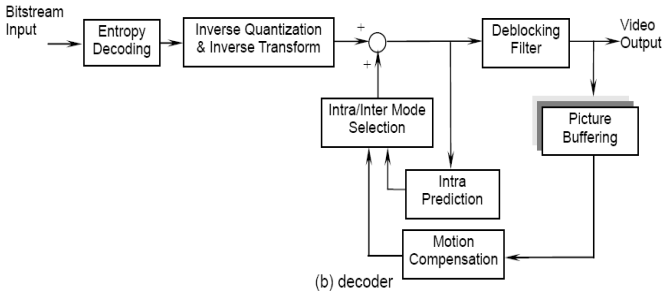


Video Coding Algorithm



(a) encoder

Video Coding Algorithm



(b) decoder

Video Coding Algorithm

- Encoder may select between **intra** and **intercoding** for block-shaped regions of each picture.
- **Intra coding**
  - can provide **access points** to the coded sequence where decoding can begin and continue correctly
  - uses **various spatial prediction** modes to reduce spatial redundancy in the source signal for a single picture
- **Inter coding (predictive or bi-predictive)**
  - more efficient using inter prediction of each block of sample values from **some previously** decoded pictures
  - uses motion vectors for block-based inter prediction to reduce temporal redundancy among different pictures. Prediction is obtained from **deblocking filtered signal of previous reconstructed pictures**. The deblocking filter is to reduce the blocking artifacts at the block boundaries. Motion vectors and intra prediction modes may be specified for a **variety of block sizes in the picture**. The prediction residual is then further compressed using a transform to remove spatial correlation in the block before it is quantized. Finally, the motion vectors or intra prediction modes are combined with the quantized transform coefficient information and encoded using entropy code such as context-adaptive variable length codes (CAVLC) or context adaptive binary arithmetic coding (CABAC).

Intra Prediction

- **Previous standards**:
  - have adopted the Intra-coded macroblock, coded by itself without temporal prediction
  - Intra-coded macroblock occurs in Intra-coded slice or the macroblock having unacceptable temporal correction of motion compensated prediction.
  - Intra-coded macroblock introduces the high amount of coded bits.

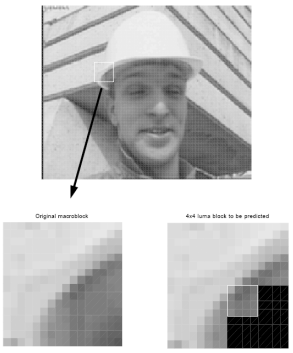
This is a **bottleneck for reducing the bitrate**.

H.264 uses the methods of **predicting intra-coded macroblocks** to reduce the high amount of bits coded by original input signal itself. For encoding a block or macroblock in Intra-coded mode, a prediction block is formed based on previously reconstructed (but, unfiltered for deblocking) blocks. The residual signal between the current block and the prediction is finally encoded.

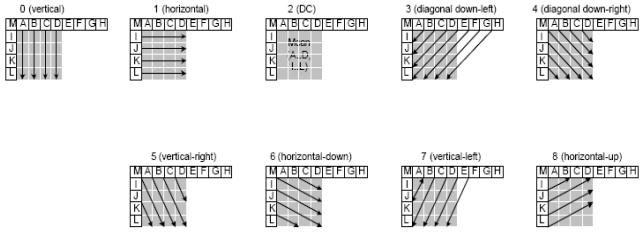
Intra Prediction Modes

	Block size	Modes
Luma prediction	4x4	9
	8x8	9
	16x16	4
Chroma prediction	8x8 (4:2:0) 8X16 (4:2:2) 16x16 (4:4:4)	4

Example 4x4 block



Luma 4x4 – 9 prediction modes



Examples:

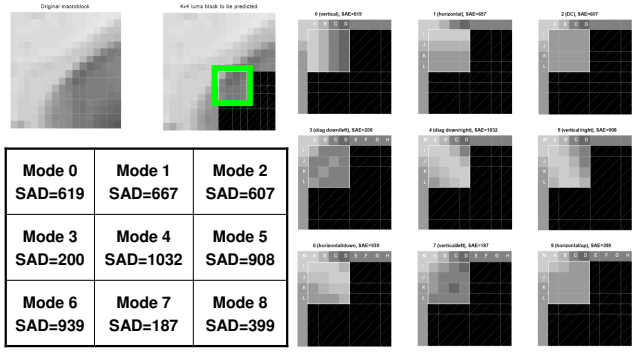
$$P(0,0) = round\left(\frac{I}{4}+\frac{M}{2}+\frac{A}{4}\right), \quad P(0,3) = round\left(\frac{B}{4}+\frac{C}{2}+\frac{D}{4}\right) \quad (\text{mode } 4)$$

Prediction modes

- For mode 0 (vertical) and mode 1 (horizontal), the predicted samples are formed by extrapolation from upper samples [A, B, C, D] and from left samples [I, J, K, L], respectively.
- For mode 2 (DC), all of the predicted samples are formed by mean of upper and left samples [A, B, C, D, I, J, K, L].
- For mode 3 (diagonal down left), mode 4 (diagonal down right), mode 5 (vertical right), mode 6 (horizontal down), mode 7 (vertical left), and mode 8 (horizontal up), the predicted samples are formed from a weighted average of the prediction samples A-M. For example, samples a and d are respectively predicted by round(I/4 + M/2 + A/4) and round(B/4 + C/2 + D/4) in mode 4, also by round(I/2 + J/2) and round(J/4 + K/2 + L/4) in mode 8.

The encoder may select the prediction mode for each block that minimizes the residual between the block to be encoded and its prediction.

4x4 Prediction Example

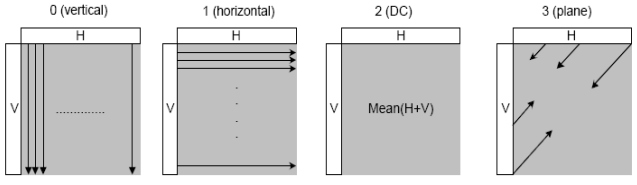


Mode 0 SAD=619	**Mode 1 SAD=667**	**Mode 2 SAD=607**
Mode 3 SAD=200	**Mode 4 SAD=1032**	**Mode 5 SAD=908**
Mode 6 SAD=939	**Mode 7 SAD=187**	**Mode 8 SAD=399**

Intra Prediction

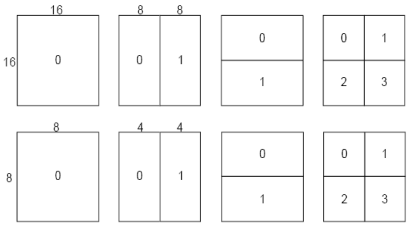
- For prediction of each 8 x 8 luma block, one mode is selected from the 9 modes, similar to the (4x4) intrablock prediction.
- For prediction of all 16 x 16 luma components of a macroblock, four modes are available.
  - For mode 0 (vertical), mode 1 (horizontal), mode 2 (DC), the predictions are similar with the cases of 4 x 4 luma block.
  - For mode 4 (Plane), a linear plane function is fitted to the upper and left samples.
- Each chroma component of a macroblock is predicted from chroma samples above and/or to the left that have previously been encoded and reconstructed. The chroma prediction is defined for three possible block sizes:
  - 8 x 8 chroma in 4:2:0 format
  - 8 x 16 chroma in 4:2:2 format
  - 16 x 16 chroma in 4:4:4 format.
- The 4 prediction modes for all of these cases are very similar to the 16 x 16 luma prediction modes, except that the order of mode numbers is different: mode 0 (DC), mode 1 (horizontal), mode 2 (vertical), and mode 3 (plane).

Luma 16x16 – 4 prediction modes

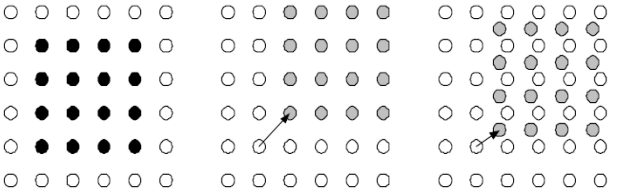


Inter Prediction

- Macroblock partitions
- Block partitions



Motion Vector Examples



Target block (4x4) **Reference v(1,-1)** **Reference v(0.75,-.5)**

Motion estimation: full-sample, half-sample, quarter-sample

From integer to half pixel

Integer-pixel

	0	1	2	3	4	5
0			A	B		
1			C	D		
2	E	F	G	H	I	J
3	K	L	M	N	P	Q
4			R	S		
5			T	U		

Half-pixel

	0	0.5	1	1.5	2	2.5	3	3.5	4	4.5	5
0					A		B				
0.5											
1					C		D				
1.5											
2	E		F		G		H		I		J
2.5											
3	K		L		M		N		P		Q
3.5											
4					R		S				
4.5											
5					T		U				

Half-pixel accuracy

				A	aa	B				
				C	bb	D				
E		F		G	b	H		I		J
cc		dd		h	j			ee		ff
K		L		M		N		P		Q
				R	gg	S				
				T	hh	U				

$$b_1 = (E - 5F + 20G + 20H - 5I + J)$$

$$h_1 = (A - 5C + 20G + 20M - 5R + T)$$

$$b = (b_1 + 16) >> 5$$

$$h = (h_1 + 16) >> 5$$

$$j_1 = (cc - 5dd + 20h_1 + 20m_1 - 5ee + ff)$$

$$j = (j_1 + 512) >> 10$$

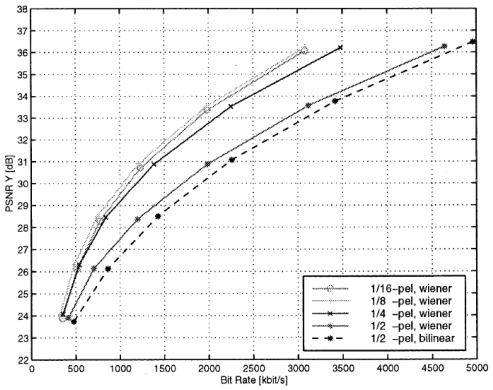
Quarter-pixel accuracy

							A		aa		B							
							C		bb		D							
E				F			G	a	b	c	H				I			J
							d	e	f	g								
cc				dd			h	i	j	k	m				ee			ff
							n	p	q	r								
K				L			M		s		N				P			Q
							R		gg		S							

$$a = (G + b + 1) >> 1, \quad d = (G + h + 1) >> 1, \quad e = (b + h + 1) >> 1$$

							T		hh		U							

Examples: RD Curves



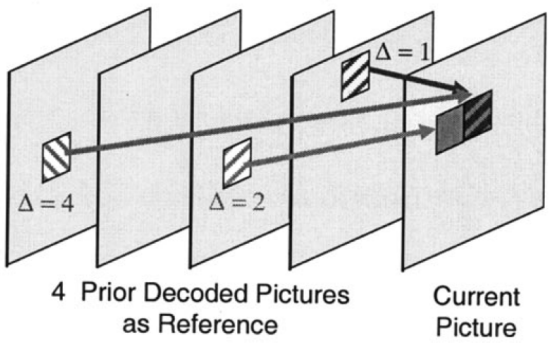
Inter Prediction

- Inter prediction is to reduce the temporal correlation with help of motion estimation and compensation. In H.264, the current picture can be partitioned into the macroblocks or the smaller blocks. A macroblock of 16x16 luma samples can be partitioned into smaller block sizes up to 4x4.
- For 16x16 macroblock mode:
  - there are 4 cases : 16x16, 16x8, 8x16 or 8x8,
- For 8x8 macroblock mode:
  - also 4 cases : 8x8, 8x4, 4x8 or 4x4 for 8x8 mode.
- The smaller block size requires larger number of bits to signal the motion vectors and extra data of the type of partition, however the motion compensated residual data can be reduced. Therefore, the choice of partition size depends on the input video characteristics. In general, a large partition size is appropriate for homogeneous areas of the frame and a small partition size may be beneficial for detailed areas.

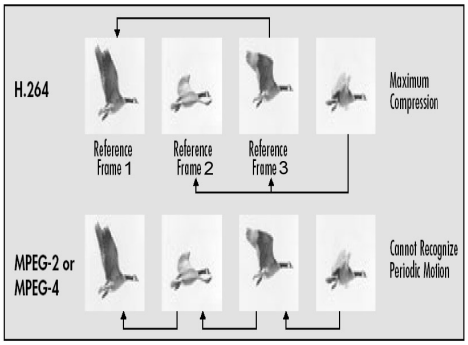
Inter Prediction

- The inter prediction process can form segmentations for motion representation as small as 4 x 4 luma samples in size, using motion vector accuracy of **one-quarter of the luma sample**. Sub-pel motion compensation can provide significantly better compression performance than integer-pel compensation, at the expense of increased complexity. Quarter-pel accuracy outperforms half-pel accuracy. Especially, sub-pel accuracy would increase the coding efficiency at high bitrates and high video resolutions.
- In the luma component, the sub-pel samples at half-pel positions are generated first and are interpolated from neighboring integer-pel samples using a 6-tap FIR filter with weights (1, -5, 20, 20, -5, 1)/32. Once all the half-pel samples are available, each quarter-pel sample is produced using bilinear interpolation between neighboring half- or integer-pel samples. For 4:2:0 video source sampling, 1/8 pel samples are required in the chroma components (corresponding to 1/4 pel samples in the luma). These samples are interpolated (linear interpolation) between integer-pel chroma samples. Sub-pel motion vectors are encoded differentially with respect to predicted values formed from nearby encoded motion vectors.

Multiframe motion compensation



Multiframe motion compensation



Inter Prediction

- The process for inter prediction of a sample block can also involve the selection of the pictures to be used as the reference pictures from a number of stored previously-decoded pictures. Reference pictures for motion compensation are stored in the picture buffer. With respect to the current picture, the pictures before and after the current picture, in the display order are stored into the picture buffer. These are classified as 'short-term' and 'long-term' reference pictures. Long-term reference pictures are introduced to extend the motion search range by using multiple decoded pictures, instead of using just one decoded short-term picture. Memory management is required to take care of marking some stored pictures as 'unused' and deciding which pictures to delete from the buffer for efficient memory management.

Transform and quantization

- Both source pictures and prediction residuals have high spatial redundancies. H.264 Standard is based on the use of a block-based transform for spatial redundancy removal. After inter prediction from previouslydecoded samples in other pictures or spatial-based prediction from previously-decoded samples within the current picture, the resulting prediction residual is split into 4 x 4 or 8 x 8 blocks. These are converted into the transform domain where they are quantized.
- H.264 uses an adaptive transform block size, 4 x 4 and 8 x 8 (High Profiles only), whereas previous video coding standards used the 8 x 8 DCT. The smaller block size leads to a significant reduction in ringing artifacts. Also, the 4 x 4 transform has the additional benefit of removing the need for multiplications.
- For improved compression efficiency, H.264 also employs a hierarchical transform structure, in which the DC coefficients of neighboring 4 x 4 transforms for the luma signals are grouped into 4 x 4 blocks and transformed again by the Hadamard transform.

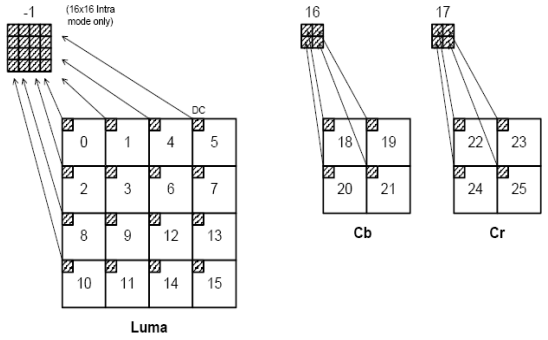
Transform and quantization

- For blocks with mostly flat pel values, there is significant correlation among transform DC coefficients of neighboring blocks. Therefore, the standard specifies the 4 x 4 Hadamard transform for luma DC coefficients for 16 x 16 Intra-mode only, and 2 x 2 Hadamard transform for chroma DC coefficients.
- In some applications, it is desired to reduce the quantization step size to improve PSNR to levels that can be considered visually lossless. To achieve this, the H.264 extends the quantization step sizes QP by two additional octaves, redefining the tables and allowing QP to vary from 0 to 51.
- In general transform and quantization require several multiplications resulting in high complexity for implementation. So, for simple implementation, the exact transform process is modified to avoid the multiplications. Then the transform and quantization are combined by the modified integer forward transform, quantization, scaling.

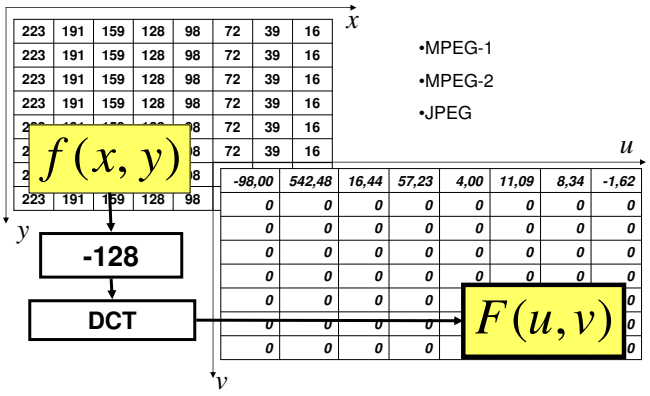
Transform and quantization

- Encoding process:
  - forward integer transform;
  - post-scaling and quantization;
- Decoding process:
  - inverse quantization and pre-scaling;
  - inverse integer transform;
- Quantization scaling matrices
  - The High Profiles support the perceptual-based quantization scaling matrices as same concept used in MPEG-2. The encoder can specify a matrix for scaling factor according to the specific frequency associated with the transform coefficient for use in inverse quantization scaling by the decoder. This allows the optimization of the subjective quality according to the sensitivity of the human visual system, less sensitive to the coded error in high frequency transform coefficients. H.264 suggests default scaling matrices.

Block order scan



DCT 8x8



Discrete Cosine Transforms

The NxN 2-dimensional DCT is defined as:

$$F(u,v) = \frac{2}{N} C(u)C(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y) \cos\frac{(2x+1)u\pi}{2N} \cos\frac{(2y+1)v\pi}{2N}$$

$$f(x,y) = \frac{2}{N} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} C(u)C(v)F(u,v) \cos\frac{(2x+1)u\pi}{2N} \cos\frac{(2y+1)v\pi}{2N}$$

$$C(u), C(v) = \begin{cases} \frac{1}{\sqrt{2}} & \text{for } u,v = 0 \\ 1 & \text{otherwise} \end{cases}$$

DCT8x8 Example code

```
int N = 8;
double f[8][8];
double F[8][8];

double C(double u)
{
   if (u == 0.0)
        return 1.0/sqrt(2.0);
   else
        return 1.0;
}
```

DCT8x8 Example code

```
void _DCTTransform(int u, int v) {
    for(int x = 0; x < N; ++x)
        for(int y = 0; y < N; ++y)
            F[u][v] += (f[x][y]-128)*
                cos((2*x+1)*u*M_PI/(2*N))*
                cos((2*y+1)*v*M_PI/(2*N));

    F[u][v] *= 2/N*C(u)*C(v);
}

void DCTTransform() {
    for(int u = 0; u < N; ++u)
        for(int v = 0; v < N; ++v) {
            F[u][v] = 0.0;
            _DCTTransform(u,v);
        }
}
```


DCT 4x4

223	191	159	128
223	191	159	128
223	191	159	128
223	191	159	128

189,00	*141,44*	*1,00*	*9,60*
0	*0*	*0*	*0*
0	*0*	*0*	*0*
0	*0*	*0*	*0*

$f(x,y)$ → **-128** → **DCT** → $F(u,v)$

$$F = \begin{bmatrix} F_{00} & F_{01} & F_{02} & F_{03} \\ . & . & . & . \\ . & . & . & . \\ . & . & . & F_{33} \end{bmatrix} = \begin{bmatrix} v_0 & v_0 & v_0 & v_0 \\ v_1 & v_1 & v_1 & v_1 \\ v_2 & v_2 & v_2 & v_2 \\ v_3 & v_3 & v_3 & v_3 \end{bmatrix} f \begin{bmatrix} u_0 & u_1 & u_2 & u_3 \\ u_0 & u_1 & u_2 & u_3 \\ u_0 & u_1 & u_2 & u_3 \\ u_0 & u_1 & u_2 & u_3 \end{bmatrix}$$

DCT 4x4

$$C(u), C(v) = \begin{cases} 1/\sqrt{2} & for\ u,v = 0 \\ 1 & otherwise \end{cases}$$

$$F(u,v) = \frac{2}{N} C(u)C(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y) \cos\frac{(2x+1)u\pi}{2N} \cos\frac{(2y+1)v\pi}{2N}$$

$$\tilde{F} = \begin{bmatrix} F_{00} & F_{01} & F_{02} & F_{03} \\ . & . & . & . \\ . & . & . & . \\ . & . & . & F_{33} \end{bmatrix} = \begin{bmatrix} v_0 & v_0 & v_0 & v_0 \\ v_1 & v_1 & v_1 & v_1 \\ v_2 & v_2 & v_2 & v_2 \\ v_3 & v_3 & v_3 & v_3 \end{bmatrix} f \begin{bmatrix} u_0 & u_1 & u_2 & u_3 \\ u_0 & u_1 & u_2 & u_3 \\ u_0 & u_1 & u_2 & u_3 \\ u_0 & u_1 & u_2 & u_3 \end{bmatrix}$$

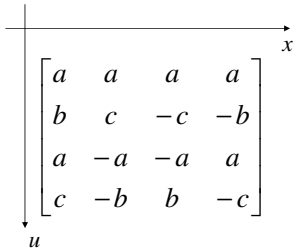
DCT 4x4

$$s(x,u) = \cos\frac{(2\cdot x\cdot+1)\cdot u\cdot\pi}{2N},\ N=4 \qquad s(0,0) = \cos\frac{(2\cdot 0\cdot+1)\cdot 0\cdot\pi}{8} = 1$$

$$\begin{array}{cccc} s(0,0)=1 & s(1,0)=1 & s(1,0)=1 & s(1,0)=1 \\ s(0,1)=\cos\left(\frac{\pi}{8}\right) & s(1,1)=\cos\left(\frac{3\pi}{8}\right) & s(2,1)=\cos\left(\frac{5\pi}{8}\right)=-\cos\left(\frac{3\pi}{8}\right) & s(3,1)=\cos\left(\frac{7\pi}{8}\right)=\cos\left(\frac{\pi}{8}\right) \\ s(0,2)=\cos\left(\frac{\pi}{8}\right) & s(1,2)=\cos\left(\frac{6\pi}{8}\right)=-1 & s(2,2)=\cos\left(\frac{10\pi}{8}\right)=1 & s(3,2)=\cos\left(\frac{14\pi}{8}\right)=-1 \end{array}$$

DCT 1-D Bases

$$H(x,u) = C(u)\sqrt{\frac{2}{N}} \cos\left[\frac{(2x+1)u\pi}{2N}\right] \qquad C(u) = \begin{cases} \frac{1}{\sqrt{2}} & for\ u=0 \\ 1 & otherwise \end{cases}$$



$$\begin{bmatrix} a & a & a & a \\ b & c & -c & -b \\ a & -a & -a & a \\ c & -b & b & -c \end{bmatrix}$$

$$a = \frac{1}{2}$$

$$b = \sqrt{\frac{1}{2}}\cos\left(\frac{\pi}{8}\right)$$

$$c = \sqrt{\frac{1}{2}}\cos\left(\frac{3\pi}{8}\right)$$

Cosines values

$$\begin{bmatrix} a & a & a & a \\ b & c & -c & -b \\ a & -a & -a & a \\ c & -b & b & -c \end{bmatrix}$$

$$a = \frac{1}{2}$$

$$b = \sqrt{\frac{1}{2}}\cos\left(\frac{\pi}{8}\right)$$

$$c = \sqrt{\frac{1}{2}}\cos\left(\frac{3\pi}{8}\right)$$

$$\begin{bmatrix} 0.5 & 0.5 & 0.5 & 0.5 \\ 0.653 & 0.271 & -0.271 & -0.653 \\ 0.5 & -0.5 & -0.5 & 0.5 \\ 0.271 & -0.653 & 0.653 & -0.271 \end{bmatrix}$$

Fixed point DCT in H.263+

$$\text{round}(\alpha H) \quad \begin{matrix} a=13 \\ b=17 \\ c=7 \end{matrix} \quad \alpha = 26 \qquad \begin{bmatrix} 13 & 13 & 13 & 13 \\ 17 & 7 & -7 & -17 \\ 13 & -13 & -13 & 13 \\ 7 & -17 & 17 & -7 \end{bmatrix}$$

- Very similar to original DCT
- Orthogonal rows
- Constant row norm
- Higher dynamic range
  - Max value of Hx = 52A

$$\log_2(52^2) = 11.4$$

Fixed point DCT in H.264

$$\text{round}(\alpha H) \quad \begin{matrix} a=1 \\ b=2 \\ c=1 \end{matrix} \quad \alpha = 2.5 \qquad \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix}$$

- Also has orthogonal rows
- Non-constant row norm

From DCT to integer transform

$$\mathbf{Y} = \mathbf{AXA}^T = \begin{bmatrix} a & a & a & a \\ b & c & -c & -b \\ a & -a & -a & a \\ c & -b & b & -c \end{bmatrix} \mathbf{X} \begin{bmatrix} a & b & a & c \\ a & c & -a & -b \\ a & -c & -a & b \\ a & -b & a & -c \end{bmatrix} \qquad \begin{matrix} a = \frac{1}{2} \\ b = \sqrt{\frac{1}{2}}\cos\left(\frac{\pi}{8}\right) \\ c = \sqrt{\frac{1}{2}}\cos\left(\frac{3\pi}{8}\right) \end{matrix}$$

$$\mathbf{Y} = \left(\mathbf{CXC}^T\right) \otimes \mathbf{E} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & d & -d & -1 \\ 1 & -1 & -1 & 1 \\ d & -1 & 1 & -d \end{bmatrix} \mathbf{X} \begin{bmatrix} 1 & 1 & 1 & d \\ 1 & d & -1 & -1 \\ 1 & -d & -1 & 1 \\ 1 & -1 & 1 & -d \end{bmatrix} \otimes \begin{bmatrix} a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \\ a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \end{bmatrix}$$

Integer transform

Approximations

$$a = \frac{1}{2} \qquad b = \sqrt{\frac{2}{5}} \qquad d = \frac{1}{2}$$

Integer transform (X_{ij})

SF_{ij}, post scaling factor

$$\mathbf{Y} = \mathbf{C}_f \mathbf{X} \mathbf{C}_f^T \otimes \mathbf{E}_f = \underbrace{\begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix}}_{H} \mathbf{X} \begin{bmatrix} 1 & 2 & 1 & 1 \\ 1 & 1 & -1 & -2 \\ 1 & -1 & -1 & 2 \\ 1 & -2 & 1 & -1 \end{bmatrix} \otimes \begin{bmatrix} a^2 & ab/2 & a^2 & ab/2 \\ ab/2 & b^2/4 & ab/2 & b^2/4 \\ a^2 & ab/2 & a^2 & ab/2 \\ ab/2 & b^2/4 & ab/2 & b^2/4 \end{bmatrix}$$

Multiplication in the transform process is avoided by integrating it with the quantization.

2D Hadamard transform

- For luma (4 x 4) DC coefficients in 16 x 16 Intra-mode, 2D Hadamard transform is applied.

$$\mathbf{Y}_D = \left(\underbrace{\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix}}_{\hat{H}} \mathbf{W}_D \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \right) / 2$$

2D Hadamard transform

For chroma DC coefficients in 4:2:0 format, the transform matrix is as follows

$$\hat{H} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

For 4:2:2 and 4:4:4 formats, the Hadamard block size is increased to reflect the enlarged blocks.

DCT 8x8 for High Profile

$$\overline{H} = \begin{bmatrix} 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 \\ 12 & 10 & 6 & 3 & -3 & -6 & -10 & -12 \\ 8 & 4 & -4 & -8 & -8 & -4 & 4 & 8 \\ 10 & -3 & -12 & -6 & 6 & 12 & 3 & -10 \\ 8 & -8 & -8 & 8 & 8 & -8 & -8 & 8 \\ 6 & -12 & 3 & 10 & 10 & -3 & 12 & -6 \\ 4 & -8 & 8 & -4 & -4 & 8 & -8 & 4 \\ 3 & -6 & 10 & -12 & 12 & -10 & 6 & -3 \end{bmatrix}$$

Quantization

QP	0	1	2	3	4	5	6	7	8	9	10	11	12	....
QStep	0.625	0.6875	0.8125	0.875	1	1.125	1.25	1.375	1.625	1.75	2	2.25	2.5	....
QP	...	18	...	24	...	30	...	36	...	42	...	48	...	51
QStep		5		10		20		40		80		160		224

$$Z_{ij} = Y_{ij} round\left(\frac{SF_{ij}}{Q_{step}}\right)$$

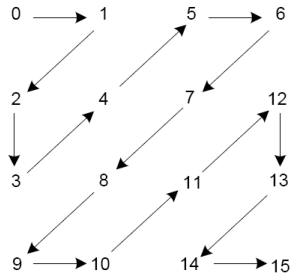
Quantization scaling matrices (High Profile)

- The High Profiles support the perceptual-based quantization scaling matrices as same concept used in MPEG-2. The encoder can specify a matrix for scaling factor according to the specific frequency associated with the transform coefficient for use in inverse quantization scaling by the decoder. This allows the optimization of the subjective quality according to the sensitivity of the human visual system, less sensitive to the coded error in high frequency transform coefficients. H.264 suggests default scaling matrices.

$$Y_{ij} = X_{ij} round\left(\frac{SF_{ij}}{Q_{step} \cdot w_{ij}}\right)$$

The default perceptual weighting matrices are suggested for 4x4 intra Y, Cb, Cr,
4x4 inter Y, Cb, Cr and 8x8 intra/inter Y.

Coefficient order



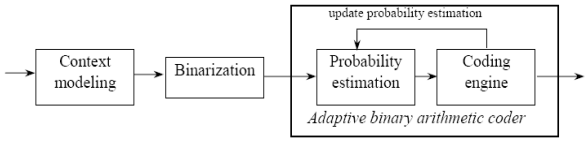
Entropy coding

- Entropy coding in previous standards such as MPEG-1, 2, 4, H.261, and H.263 is based on **fixed tables of variable length codes (VLCs)**. These standards define sets of codewords based on the probability distributions of generic videos instead of exact Huffman code for the video sequences. However H.264 uses different VLCs in order to match a symbol to a code based on the context characteristics.
- All syntax elements except for the residual data are encoded by the Exp-Golomb codes. In order to read the residual data (quantized transform coefficients), zig-zag scan (interlaced) or alternate scan (noninterlaced or field) is used.
- For coding the residual data, a more sophistical method called CAVLC is employed. Also, CABAC is employed in Main and High profiles, CABAC has more coding efficiency but higher complexity compared to CAVLC.

CAVCL and CABAC

- Context-based Adaptive Variable Length Coding (CAVLC)
  - After transform and quantization, the probability that the level of coefficients is zero or +/-1 is very high.
  - CAVLC handles the zero and +/-1 coefficients as the different manner with the levels of coefficients. The total numbers of zero and +/-1 are coded. For other coefficients, their levels are coded.
- Context-based Adaptive Binary Arithmetic Coding (CABAC)
  - CABAC utilizes the arithmetic coding, also in order to achieve good compression, the probability model for each symbol element is updated as shown in the following Figure. The CABAC encoding process consists of three elementary steps.

CABAC



- Step 1: binarization
  - A given nonbinary valued symbol (e.g. a transform coefficient or motion vector) is uniquely mapped to a binary sequence prior to arithmetic coding. This process is similar to the process of converting a data symbol into a variable length code but the binary code is further encoded by the arithmetic coder prior to transmission.
- Step 2: context modeling
  - A context model is a probability model for one or more elements of the binarized symbol. The probability model is selected such that the corresponding choice may depend on previously encoded syntax elements.
- Step 3: binary arithmetic coding
  - An arithmetic coder encodes each element according to the selected probability model together with a subsequent model updating.

Entropy Coding

- Parameters that require to be encoded and transmitted
  - Macroblock Type (Prediction method)
  - QP
  - Motion data: Reference frame and Motion Vector
  - Coded block pattern (blocks containing coded coefficients)
  - Residual Data
- H.264/MPEG-4 AVC uses a number of techniques for entropy coding:
  - Exp-Golomb codes: all syntax elements except the quantized transform coefficients.
  - Context Adaptive Variable Length Coding (CAVLC): Quantized transform coefficients
  - Context Adaptive Binary Arithmetic Coding (CABAC) : Quantized transform coefficients

Exp-Golomb Codes

- Exp-Golomb codes (Exponential Golomb codes) are variable length codes with a regular construction:
  - uses a single infinite-extent codeword table
  - only the mapping to the single codeword table is customized according to the data statistics

Exp-Golomb Codeword

[M zeros][1][INFO]

codeword length → (2M+1) bits

code_num	Codeword
0	1
1	010
2	011
3	00100
4	00101
5	00110
6	00111
7	0001000
8	0001001
…	…

Encoding

$M = \text{round}[\log_2(\text{code\_num}+1)]$

$INFO = \text{code\_num}+1-2^M$

Decoding

$\text{Code\_num} = 2^M + INFO - 1$

Mapping

v	code_num
0	0
1	1
2	2
3	3
4	4
5	5
…	…

- Each element **v** is assigned a type reflecting how data is to be mapped
- Unsigned Exponential → ue(v)
  - Mapping: code_num = v
  - Used for macroblock type, reference frame index
- Signed Exponential → se(v)
  - Mapping
    - code_num = 2|v| (v < 0)
    - code_num = 2|v| - 1 (v > 0)
  - Used for motion vector difference, QP.
- Mapped Exponential → me(v)
  - Mapping specified in the standard
  - Used for the Coded block patterns:
    - 8x8 blocks containing non-zero coefficients

coded_block_pattern (Inter prediction)	code_num
0 (no non-zero blocks)	0
16 (chroma DC block non-zero)	1
1 (top-left 8x8 luma block non-zero)	2
2 (top-right 8x8 luma block non-zero)	3
4 (lower-left 8x8 luma block non-zero)	4
8 (lower-right 8x8 luma block non-zero)	5
32 (chroma DC and AC blocks non-zero)	6
3 (top-left and top-right 8x8 luma blocks non-zero)	7
…	…

- Truncated Exponential → te(v)
- Each mapping is designed to produce short codewords for frequently occurring values and longer codewords for less common parameter values.

CAVLC

- Method for coding residual, zig-zag ordered transformed blocks
- VLC tables for various syntax elements are switched depending on already transmitted syntax elements
- CAVLC takes advantage of several characteristics of quantized 4x4 blocks:
  - Sparse blocks (containing mostly zeros)
  - Highest non-zero coefficients after the zig-zag scan are often sequences of +/-1 (“Trailing 1s” or “T1s”)
  - The number of non-zero coefficients in neighbouring blocks is correlated
  - The level (magnitude) of non-zero coefficients tends to be higher at the start of the reordered array (near the DC coefficient) and lower towards the higher frequencies

CAVLC encoding

- CAVLC encoding of a block of transform coefficients proceeds as follows:
  - Encode the number of coefficients and trailing ones in a 4x4 block
  - Encode the sign of each T1
  - Encode the levels of the remaining non-zero coefficients
  - Encode the total number of zeros before the last coefficient
  - Encode each run of zeros

CAVLC Step 1

- Encode the number of coefficients (*TotalCoeffs*) and T1s
  - TotalCoeffs range from 0 to 16 (in a 4x4 block)
  - T1s range from 0 to 3 (max value allowed)
- Coding: 4 choices for look-up table (*coeff_token*)
  - Num_VCL0 → biased towards small numbers of coefficients
  - Num_VCL1 → biased towards medium numbers of coefficients
  - Num_VCL2 → biased towards higher numbers of coefficients
  - Num_FLC → fixed 6-bit length code
- The choice depends on the number of non-zero coefficients in upper and left-hand previously coded blocks NU and NL (context adaptivity)

CAVLC Step 1

- U,L available → N=(NU+NL)/2
- U available → N=NU
- L available → N=NL
- U,L unavailable → N=0

N	Table
0, 1	Num-VLC0
2, 3	Num-VLC1
4, 5, 6, 7	Num-VLC2
8 or above	FLC

CAVLC Step 2

- Step 2
  - Encode the sign of each T1.
  - For each T1 up to three, a single bit encodes the sign (0=+, 1=-). Encoding is in reverse order, starting with the highest-frequency T1

CAVLC Step 3

- Step 3
  - Encode the levels of the remaining non-zero coefficients.
  - The level (sign and magnitude) of each remaining non-zero coefficient in the block is encoded in reverse order
  - Encoding VLC table depends on successive coded level (context adaptivity)

1. Initialise the table to Level_VLC0
2. Encode highest-frequency coefficient
3. If magnitude is larger than a threshold, move up to the next VLC table

Current VLC table	Threshold to increment table
VLC0	0
VLC1	3
VLC2	6
VLC3	12
VLC4	24
VLC5	48
VLC6	N/A (highest table)

CAVLC Step 4

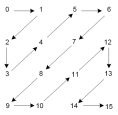
- Encode the total number of zeros before the last coefficient
- Coding with a VLC of the number of all zeros preceding the highest non-zero coefficient.

CAVLC Step 5

- Encode each run of zeros
- The number of zeros preceding each non-zero coefficient (*run_before*) is encoded in reverse order
- The VLC for each run of zeros is chosen depending on
  - the number of zeros that have not yet been encoded (*ZerosLeft*)
  - value of *run_before* parameter.

Example Encoding CAVLC

0	3	-1	0
0	-1	1	0
1	0	0	0
0	0	0	0



	Element	Value	Code
1	coeff_token	TotalCoeffs=5, T1s=3	0000100
2	T1 sign (4)	+	0
	T1 sign (3)	-	1
	T1 sign (2)	-	1
3	Level (1)	+1 (use Level_VLC0)	1
	Level (0)	+3 (use Level_VLC1)	0010
4	TotalZeros	3	111
5	run_before(4)	ZerosLeft=3; run_before=1	10
	run_before(3)	ZerosLeft=2; run_before=0	1
	run_before(2)	ZerosLeft=2; run_before=0	1
	run_before(1)	ZerosLeft=2; run_before=1	01
	run_before(0)	ZerosLeft=1; run_before=1	No code required; last coefficient.

- Reordered Block: 0,3,0,1,-1,-1,0,1,0,0,0,0,0,0,0,0
- *TotalCoeff* = 5; *TotalZeros*=3; T1s =3 (max value)
- Bitstream → 000010001110010111101101 (24 bits)

Example Decoding CAVLC

- Bitstream → 000010001110010111101101 (24 bits)

Code	Element	Value	Output array
0000100	coeff_token	TotalCoeffs=5, T1s=3	Empty
0	T1 sign	+	1
1	T1 sign	-	-1, 1
1	T1 sign	-	-1, -1, 1
1	Level	+1	1, -1, -1, 1
0010	Level	+3	3, 1, -1, -1, 1
111	TotalZeros	3	3, 1, -1, -1, 1
10	run_before	1	3, 1, -1, -1, 0, 1
1	run_before	0	3, 1, -1, -1, 0, 1
1	run_before	0	3, 1, -1, -1, 0, 1
01	run_before	1	3, 0, 1, -1, -1, 0, 1

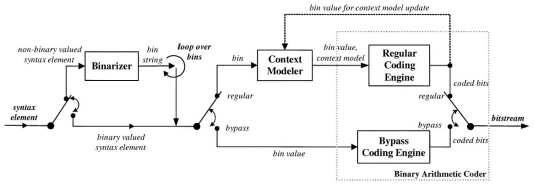
The decoder has inserted two zeros; however, TotalZeros is equal to 3 and so another 1 zero is inserted before the lowest coefficient, making the final output array: 0,3,0,1,-1,-1,0,1,…

CABAC

- Arithmetic coding makes effective use of probability models of occurrence of symbols
- Particularly beneficial for symbol probabilities greater than 0.5
- The use of adaptive codes permits adaptation to non-stationary symbol statistics easily adapt to changing statistical characteristics of the data to be coded
- Context modelling: The statistics of already-coded syntax elements are used to estimate the conditional probabilities
- CABAC provides a reduction in bit-rate between 5% to 15% over CAVLC, when coding TV signals at the same quality.

CABAC basis

- Encoding with CABAC consists of three stages
  - Binarization: needed for syntax elements that are non-binary valued
  - Context modeling: a model is selected such that the choice may depend on previous encoded syntax elements or bins
  - Adaptive binary arithmetic coding: coding of bin sequence with updating of probabilities takes place

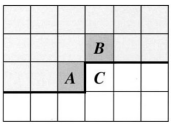


CABAC binarization

- Requirements for a successful context coding method
  - fast and accurate estimation of conditional probabilities
  - the computational complexity must be kept at a minimum
- "Pre-processing" step to reduce the alphabet size of the syntax elements
- The design of binarization schemes in CABAC (mostly) relies on a few basic code trees
  - Unary or Truncated Unary code
    - x>0 → x "1" bits plus a terminating "0" bit (unary)
    - 0<x<S→ for x=S the terminating "0" bit is neglected (truncated)
  - Exp-Golomb code: codes are constructed by a concatenation of a prefix and a suffix code word
  - Fixed-length code: a finite alphabet of values of the corresponding syntax element is assumed.

Context Modeling

- A model probability distribution is assigned to the given symbols, to drive the actual coding engine to generate a sequence of bits as a coded representation of the symbols
  - Define a modeling function $F:T \rightarrow C$ operating on a set T of past symbols
  - For each symbol x to be coded, a conditional probability $p(x|F(z))$ is estimated according to the already coded neighboring symbols
  - After encoding x, the probability model is updated with the value of the encoded symbol x



Context template consisting of two neighboring syntax elements A and B to the left and on top of the current syntax element C

B Slice

- Bidirectional prediction is very efficient to reduce the temporal correlation by using more reference pictures. Existing standards with B pictures utilize the bidirectional mode, which only allows the combinationof a previous and subsequent prediction signals. One prediction signal is derived from the subsequent inter picture, another from a previous picture, the other from a linear averaged signal of two motion compensated prediction signals.
- This H.264 generalizes this concept and supports not only forward/backward prediction pair, but also forward/forward and backward/backward pairs. Two forward references can be beneficial for motion compensated prediction of a region just before scene change, and two backward references just after scene change. **In contrast to some other previous standards, bi-directionally predictive-coded slice may also be used as references for inter coding of other pictures.**
- Also H.264 introduces the direct-mode which does not require such side information but derives reference picture, block size, and motion vector data from the subsequent inter picture. Weighted prediction is also added for the gradual transitions from scene to scene.

Weighted prediction

- All existing standards consider equal weights for reference pictures, i.e., a prediction signal is obtained by averaging with equal weights of reference signals. But, the gradual transitions from scene to scene need the different weights. The gradual transition is very popular in movies, a fade to black scene transition ('fade to black' : the luma samples of the scene gradually approach zero and the chroma samples of the scene gradually approach 128), a fade from black scene transition ('fade from black').
- H.264 uses weighted prediction method for a macroblock of P slice or B slice. A prediction signal p for B slice is obtained by different weights from two reference signals, r1 and r2.

Weighted prediction

- P = w1 x r1 + w2 x r2
- where w1 and w2 are weights. These are differently determined according to two types, explicit and implicit, in encoder. For explicit, the factors are transmitted in the slice header. For implicit, the factors are calculated based on the temporal distance between the pictures. The smaller weight is applied if the temporal distance between the reference and current pictures is close and the larger weight for the temporally long distance.

Deblocking Filter

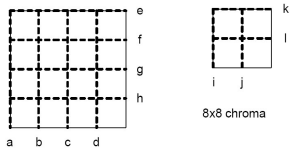
- H.264 may suffer from blocking artifacts due to block-based transform in intra and inter prediction coding, and the quantization of the transform coefficients. The deblocking filter reduces the blocking artifacts in the block boundary and prevents the propagation of accumulated coded noise. H.261 has suggested similar deblocking filter (optional) which was beneficial to reduce the temporal propagation of coded noise because only integer-pel accuracy motion compensation did not play the role for its reduction. However, MPEG-1, 2 did not use the deblocking filter because of high implementation complexity, on the other hand, the blocking artifacts can be reduced by utilizing the half-pel accuracy MC. The half-pels obtained by bilinear filtering of neighboring integer-pels played the role of the smoothing of the coded noise in the integer-pel domain.

Deblocking Filter

- H.264 uses the deblocking filter for higher coding performance in spite of implementation complexity. Filtering is applied to horizontal or vertical edges of 4 x 4 blocks in a macroblock. The luma deblocking filter process is performed on four 16-sample edges and the deblocking filter process for each chroma components is performed on two 8-sample edges.
- The deblocking filter is applied adaptively at several levels:
  - slice level: the global filtering strength can be adjusted to the individual characteristics of the video sequence.
  - block-edge level: filtering strength is made dependent on the inter/intra prediction decision, motion differences, and the presence of coded residuals in the two participating blocks. Special strong filtering is applied for macroblocks with very flat characteristics to remove 'tilting artifacts'.
  - sample level: sample values and quantizer-dependent thresholds can turn off filtering for each individual sample.

Deblocking Filter

- Benefits of the application to every decoded macroblock:
  - reduced blocking distortion block → improved image appearance
  - better motion-compensated prediction of further frames
- Filtering is applied to vertical or horizontal edges of 4x4 blocks:
  - Filter 4 vertical boundaries of the luma component (a,b,c,d)
  - Filter 4 horizontal boundaries of the luma component (e,f,g,h)
  - Filter 2 vertical boundaries of each chroma component (i,j)
  - Filter 2 horizontal boundaries of each chroma component (k,l)

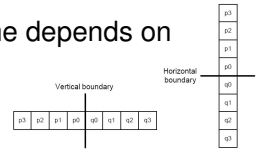


Boundary filtering: 16x16 luma

8x8 chroma

Filtering choice

- The choice of filtering outcome depends on
  - boundary strength
  - gradient across the boundary

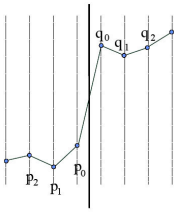


The filter is "stronger" where there is likely to be significant blocking distortion (high values of BS)

BS	Rule
4	p or q is intra coded **and** boundary **is** a macroblock boundary
3	p or q is intra coded **and** boundary **is not** a macroblock boundary
2	neither p or q is intra coded; p or q contain coded coefficients
1	neither p or q is intra coded; neither p or q contain coded coefficients; p and q have different reference frames **or** a different number of reference frames **or** different motion vector values
0	neither p or q is intra coded; neither p or q contain coded coefficients; p and q have same reference frame **and** identical motion vectors → no filtering

Filter decision

- A group of samples is filtered only if BS > 0 and the following conditions is satisfied:
  - $|p0-q0| < \alpha(QP)$
  - $|p1-p0| < \beta(QP)$
  - $|q1-q0| < \beta(QP)$ with $\beta(QP) < \alpha(QP)$
- Small QP: anything other than a very small gradient across the boundary is likely to be due to image features → low $\alpha(QP)$ and $\beta(QP)$ value
- Large QP: blocking distortion is likely to be more significant → high $\alpha(QP)$ and $\beta(QP)$ value, so that more filtering takes place.



4x4 Block Edge

Filter implementation

- BS=1,2,3
  - A 4-tap linear filter is applied with inputs p1, p0, q0 and q1, producing filtered outputs P0 and Q0
  - For luma only
    - If $|p2-p0| < \beta(QP)$, a 4-tap linear filter is applied with inputs p2, p1, p0 and q0, producing filtered output P1
    - If $|q2-q0| < \beta(QP)$, a 4-tap linear filter is applied with inputs q2, q1, q0 and p0, producing filtered output Q1

Filter implementation

- BS=4
  - If $|p2-p0| < \beta(QP)$ and $|p0-q0| < \alpha(QP)/4$
    - P0 is produced by 5-tap filtering of p2, p1, p0, q0 and q1
    - P1 is produced by 4-tap filtering of p2, p1, p0 and q0
    - Luma → P2 is produced by 5-tap filtering of p3, p2, p1, p0 and q0
  - else:
    - P0 is produced by 3-tap filtering of p1, p0 and q1
  - (the same for qi pixels)

Example of deblocking

Original Frame

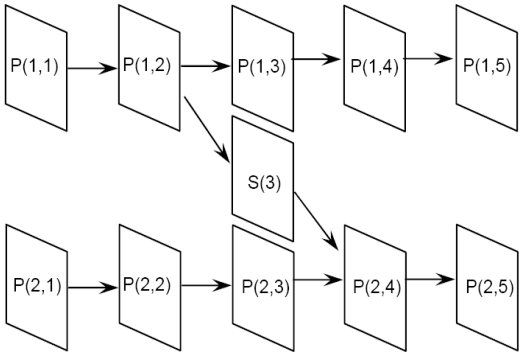
Reconstructed Frame; QP=32; No Filter

Reconstructed Frame; QP=32; With Filter

Reconstructed Frame; QP=36; No Filter

Reconstructed Frame; QP=36; With Filter

SP and SI Slices



SP and SI Slices

- In the previous standards, perfect switching between bitstreams is possible only at I picture. Reconstructing I pictures at fixed intervals allows the random access or fast playback. However, the drawback of using I picture is that it requires large number of bits, since I pictures do not exploit any temporal redundancies.
- H.264 introduces the switched slices, SP and SI, for bitstream switching.
- Figure in previous slide shows an example how to utilize SP pictures to switch between different bitstreams. Assume that there are two bitstreams, P(1,k) and P(2,k), corresponding to the same sequence encoded at different bit rates. Within each encoded bitstream, SP-pictures are placed at the locations at which switching from one bitstream to another will be allowed. In the case of switching from above bitstream P(1,3) to P(2,3), a SP picture S(3) allows to produce the decoded picture P(2,3) by using P(1,2) in the other bitstream even though motion compensations are included. SI slice, is used in a way similar to SP slice, but prediction is formed by using the 4 x 4 Intra prediction modes from previously decoded samples of the reconstructed picture.

High Fidelity Coding

- H.264 defines the special coding schemes for fidelity range extension in High Profiles. There are lossless coding schemes and support of several color formats.
- Lossless coding:
  - In order to represent the video signal as high fidelity, H.264 specifies the lossless coding schemes in High 4:4:4 Profile only. First one is PCM scheme, in which the values of the samples are sent directly for perfectly lossless representation – without prediction, transformation, or quantization. Second one is transform-bypass lossless coding scheme, which uses prediction and entropy coding of encoding sample values for fairly efficient lossless representation. Second scheme introduces less coded data than first one.

High Fidelity Coding

- Support of several color formats:
  - By performing color transformation of RGB-to-YCbCr, rounding error is introduced in both the forward and inverse color transformations. Therefore, in order to eliminate the rounding error induced in floating point operations, H.264 adds support for a new color space YCgCo as well as RGB in High Profiles.

High Fidelity Coding

$$Y = \frac{1}{2}\left(G + \frac{R+B}{2}\right), C_g = \frac{1}{2}\left(G - \frac{R+B}{2}\right), C_O = \frac{R-B}{2}$$

- To reduce the bit expansion of sample accuracy to 1 bit, the following equations are used.

$$C_O = R - B$$

$$C_g = G - (B + C_O >> 1)$$

$$Y = (B + C_O >> 1) + (C_g >> 1)$$

Error Resilience

- The previous standards use some methods to exploit the error resilience from transmission noise. The data partitioning is popular. The data are partitioned according to the significance in the bitstream, the important data then is transmitted with high priority. Also, the layered (scalable) coding induces the error resilience.
- Spatial or temporal scalable coding can recover the lost data from other layers.
- H.264 also has the error resilience property with help of S slice, parameter setting, flexible macroblock ordering, and redundant slice.

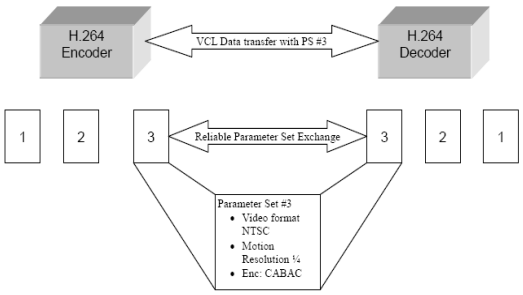
Parameter setting

- The sequence parameter set contains all information related to a sequence of pictures and a picture parameter set contains all information related to all the slices belonging to a single picture. The encoder chooses the appropriate picture parameter set to use by referencing the storage location in the slice header of each coded slice. The intelligent use of the parameter set mechanism greatly enhances error resilience.
- The key to using parameter sets in an error-prone environment is to ensure that they arrive reliably, and in a timely fashion at the decoder. They can, for example, be sent out-of-band as shown in the following Figure, using a reliable control protocol, so that the control protocol time to get them to the decoder before the relevant slices arrive over the real-time communication channel.

Parameter setting

- Alternatively, they can be sent in-band, but with appropriate application layer protection (e.g., by sending multiple copies, so as to enhance the probability that at least one copy arrives at the destination). A third option is that an application hard-codes a few parameter sets in both encoder and decoder, which would be the only operation points of the codec.

ErrRes: Parameter setting



Flexible macroblock ordering

- Flexible macroblock ordering allows assigning macroblocks to slices in an order other than the scan order. To do so, each macroblock is statically assigned to a slice group using a macroblock allocation map. Within a slice group, macroblocks are coded using the normal scan order.
- Assume that all macroblocks of the picture are allocated either to slice group 0 or slice group 1, and the macroblocks in each slice group are dispersed throughout the picture. If the packet containing the information of slice group 1 is lost during transmission, then the lost macroblock has several spatial neighbors that belong to the other slice.

Redundant slice

- Redundant slices allow to place one or more redundant representations of the same macroblocks into the same bitstream, in addition to the coded macroblocks of the slice itself. The redundant representation can be coded using different coding parameters.
- For example, the primary representation can be coded with a low quantization parameter (hence in good quality), whereas the redundant slice can be coded with a high quantization parameter (hence, in a much coarser quality, but also utilizing fewer bits). A decoder reacts to redundant slices by reconstructing only the primary slice, if it is available, and discarding the redundant slice. However, if the primary slice is missing, the redundant slice can be reconstructed.

Comparison of Coding Schemes with Other Standards

Feature/Standard	MPEG-2	MPEG-4 part 2	MPEG-4 part 10 /H.264
Macroblock size	16x16 (frame mode) 16x8 (field mode)	16x16	16x16
Block size	8x8	16x16, 16x8, 8x8	16x16, 8x16, 16x8, 8x8, 4x8, 8x4, 4x4
Intra prediction	No	Transform Domain	Spatial Domain
Transform	8x8 DCT	8x8 DCT/Wavelet transform	8x8, 4x4 integer DCT 4x4, 2x2 Hadamard
Quantization	Scalar quantization with step size of constant increment	Vector quantization	Scalar quantization with step size of increase at the rate of 12.5%
Entropy coding	VLC	VLC	VLC, CAVLC, CABAC
Pel accuracy	½-pel	¼-pel	¼ -pel
Reference picture	One picture	One picture	Multiple pictures

Comparison of Coding Schemes with Other Standards

Bidirectional prediction mode	forward / backward	forward / backward	forward / backward forward / forward backward / backward
Weighted prediction	No	No	Yes
Deblocking filter	No	No	Yes
Picture types	I, P, B	I, P, B	I, P, B, SI, SP
Profiles	5 profiles	8 profiles	7 profiles
Playback & Random access	Yes	Yes	Yes

Comparison of Coding Schemes with Other Standards

Error robustness	Data partitioning, FEC for important packet transmission	Synchronization, Data partitioning, Header extension, Reversible VLCs	Data partitioning, Parameter setting, Flexible macroblock ordering, Redundant slice, SP and SI slices
Transmission rate	2-15Mbps	64kbps - 2Mbps	64kbps - 150Mbps
Encoder complexity	Medium	Medium	High
Compatibility with previous standards	Yes	Yes	No

Comparison of Coding Schemes with Other Standards

Feature/Standard	MPEG-4 part 10 /H.264	WMV-9	AVS
Prediction block size	16x16, 8x16, 16x8, 8x8, 4x8, 8x4, 4x4	16x16, 8x8	16x16, 8x8
Intra prediction	4x4, 8x8 : 9 modes 16x16 : 4 modes	No	8x8 : 5 modes
Transform	8x8, 4x4 integer DCT 4x4, 2x2 Hadamard	8x8, 8x4, 4x8, 4x4 integer DCT	Asymmetric 8x8 integer DCT
Quantization	Scalar quantization	Dead zone, uniform scalar quantization	Scalar quantization
Entropy coding	VLC, CAVLC, CABAC	Multiple VLC tables	VLC
Sub-pel filter	1/2-pel : 6-tap 1/4-pel : 2-tap	1/2-pel : 4-tap 1/4-pel : 4-tap	1/2-pel : 4-tap 1/4-pel : 4-tap

Nota: AVS, Audio and Video Coding Standard Workgroup of China

Comparison of Coding Schemes with Other Standards

Reference picture	Multiple pictures	One picture	Two pictures
Bidirectional prediction mode	forward / backward forward / forward backward / backward 2 motion vectors	forward / backward 2 motion vectors	forward / backward symmetric 1 motion vector
Weighted prediction	Yes	Yes	Yes
Deblocking filter	Yes	Yes	Yes

Comparison of coding efficiency

Comparison of H.264 BP (baseline profile) and MPEG-4 Part2 SP (simple profile) for the MD Baseline test.
E.g. 2x at 48Kbps indicates that H.264 obtained double quality compared to MPEG4-2

Sequence	Bitrate[kbps] for QCIF				Bitrate[kbps] for CIF			
	24	48	96	192	96	192	384	768
Foreman	> 1x	2x	2x	T	2x	> 2x	T	T
Paris	> 1x	2x	2x	-	2x	2x	T, 2x	T
Head		> 2x	2x	-	2x	-	T	T
Zoom	> 1x	1x	2x	-	2x	-	-	-

-=statistically inconclusive test, T=transparency

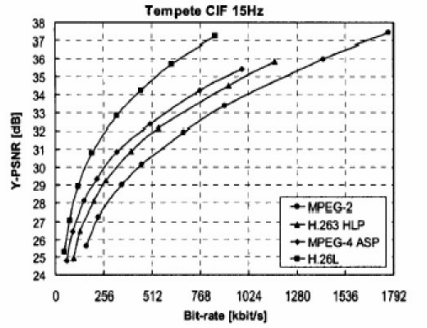
Comparison of coding efficiency

Comparison of H.264 MP (main profile) and MPEG-4 Part2 ASP (advanced simple profile) for the MD Baseline test.
E.g. 2x at 48Kbps indicates that H.264 obtained double quality compared to MPEG4-2

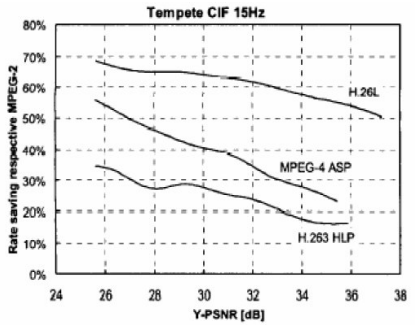
Sequence	Bitrate[kbps] for QCIF				Bitrate[kbps] for CIF			
	24	48	96	192	96	192	384	768
Football	2x / 1x	2x	2x	-	> 1x	> 1x	1x	> 1x
Mobile	2x / 1x	2x	2x	-	> 2x	4x	> 2x	T
Husky	2x	2x	> 1x	-	2x	2x	2x	-
Tempete	2x	2x	> 2x	T	2x	2x	T,2x	T

-=statistically inconclusive test, T=transparency

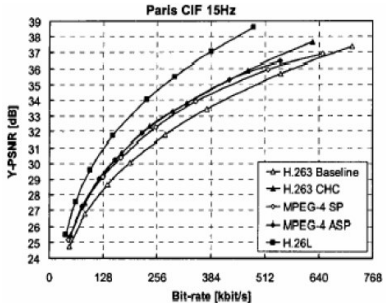
PSNR Comparison



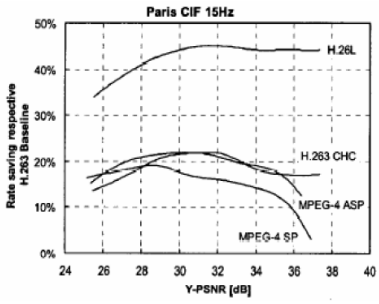
Rate saving respect MPEG-2



PSNR Comparison

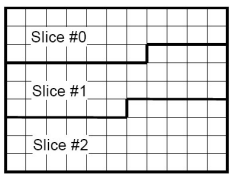


PSNR Comparison



Slice

- Ogni **picture** è partizionata in slice.
- Ogni **slice** è formata da macroblocchi.
- Ogni **macroblocco** è formato da 16x16 campioni di luminanza con i corrispondenti campioni di crominanza.



Gerarchia della sequenza video

Sequence
Pictures
Slices
Macroblocks
Macroblock partitions
Sub-macroblock partitions
Blocks
Samples (pixels)

NAL

H.264/AVC	
	VCL Video Coding Layer (Elementary Stream)
	NAL Network Abstraction Layer
Transport Layer RTP/IP, ISO MP4, H.32x, MPEG-2, ecc.	

- VLC, codifica
- NAL, incapsula i dati VLC ai fini della trasmissione su reti a pacchetto o per il multiplexing

NAL Unit

- I dati video codificati (p.e. picture slice, parameter set) sono inviati dal VCL al NAL e incapsulati in unità formate da un numero intero di byte chiamate NAL units
- Le NAL unit sono poi inviate al transport layer

NAL Unit

- Formato Byte-Stream
  - Aggiunge Start-Code-Prefix di 3 byte
  - Per sistemi H.362, MPEG-2/H222.0, ecc.
- Formato Packet-Transport
  - Lo Start-Code-Prefix non serve se il sistema ti trasporto è a pacchetti
  - Per sistemi IP/RTP.
- VCL NAL, queste unità contengono i dati dei campioni video codificati
- Non-VCL NAL, contengono le informazione associate alla codifica video (PPS, SPS, SEI, …)

NAL: byte-stream and packet-based

"The VCL is specified to efficiently represent the content of the video data. The NAL is specified to format that data and provide header information in a manner appropriate for conveyance on a variety of communication channels or storage media. All data are contained in NAL units, each of which contains an integer number of bytes. A NAL unit specifies a generic format for use in both packet-oriented and bitstream systems. The format of NAL units for both packet-oriented transport and byte stream is identical except that each NAL unit can be preceded by a start code prefix and extra padding bytes in the byte stream format."

From paragraph 7.4.1 of ITU-T Rec. H.264

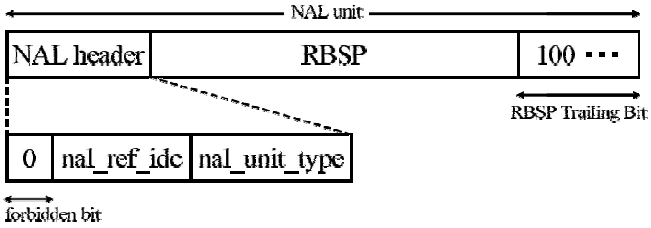
Byte stream NAL unit syntax

byte_stream_nal_unit(NumBytesInNALunit) {	C	Descriptor
while(next_bits(24) != 0x000001 && next_bits(32) != 0x00000001)		
leading_zero_8bits /* equal to 0x00 */		f(8)
if(next_bits(24) != 0x000001)		
zero_byte /* equal to 0x00 */		f(8)
start_code_prefix_one_3bytes /* equal to 0x000001 */		f(24)
nal_unit(NumBytesInNALunit)		
while(more_data_in_byte_stream( ) && next_bits(24) != 0x000001 && next_bits(32) != 0x00000001)		
trailing_zero_8bits /* equal to 0x00 */		f(8)
}		

NAL: byte-stream and packet-based

- Applicazioni per trasmissioni a pacchetto possono utilizzare direttamente le unità NAL
- Sistemi per trasmissione di flussi di bit/byte possono utilizzare la versione byte-stream delle unità NAL (ovvere le NAL precedute dallo start code)

Formato della NAL unit

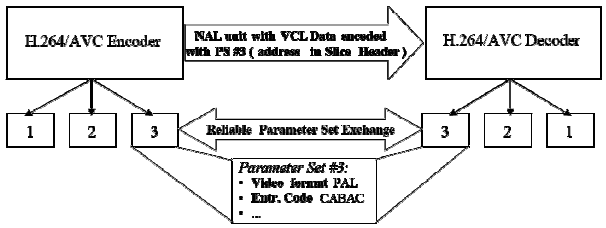


- NAL header, 1-byte
- RBSP, Raw Byte Sequence Payload di lunghezza variabile (p.e. picture slice, parameter set)

Formato della NAL unit

- NAL header:
  - forbidden bit (1 bit a 0)
  - nal_ref_idc (2 bits), indica se l'unità NAL è usata per la predizione
  - nal_unit_type (5 bits), tipo dell'unità NAL
  - payload trailing bis, aggiustano la lunghezza del payload affinché diventi multipla di byte (10…0)

Parameter Set (out-of-band)

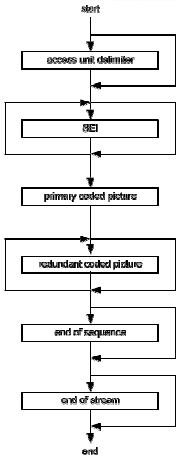


Parameter Set

- PS, informazioni che cambiano con bassa frequenza:
  - **sequence parameter sets**: relative to a series of consecutive coded video pictures (coded video sequence)
  - **picture parameter sets**: relative to one or more individual pictures.
- I PS possono essere inviati:
  - One time (ahead the VCL NAL Units)
  - Many time (to provide robustness)
  - In-band (same VCL NAL Unit Channel)
  - Out-of-Band (different Channel)

Access Units

- Insieme di unità NAL
- La decodifica di una access unit produce una picture
- Struttura:
  - **access unit delimiter**:
    - to aid in locating the start of the access unit.
  - **supplemental enhancement information (SEI)**
    - containing data such as picture timing information
  - **primary coded picture**
    - set of VCL NAL units that represent the samples of the video picture.
  - **redundant coded pictures**
    - for use by a decoder in recovering from loss or corruption
  - **end of sequence**
    - if the coded picture is the last picture of a *coded video sequence*
  - **end of stream**
    - if the coded picture is the last coded picture in the entire NAL unit stream



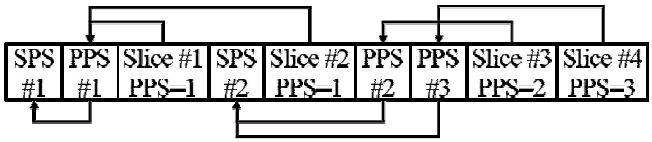
Coded Video Sequences

- Serie di access unit sequenziali
- Contiene un solo SPS
- Indipendente nella codifica rispetto a ogni altra coded-video-sequence (noto l'SPS)
- Inizia con un IDR (instantaneous decoding refresh) access unit

IDR

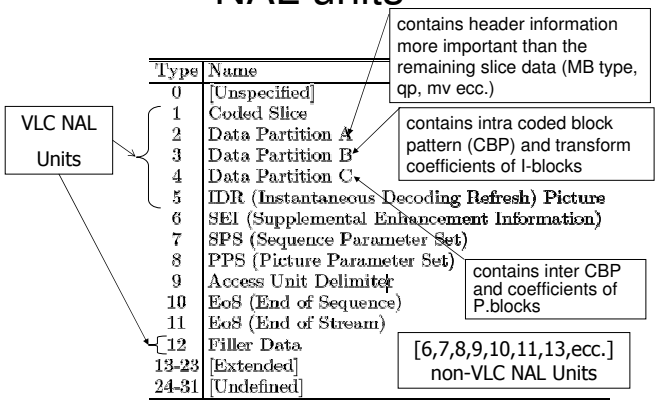
- Instantaneous decoding refresh
  - Intra frame (Frame I)
- Nessun frame che segue un'IDR può essere predetto da frame che precedeno l'IDR.

Relazione tra PPS e Slice

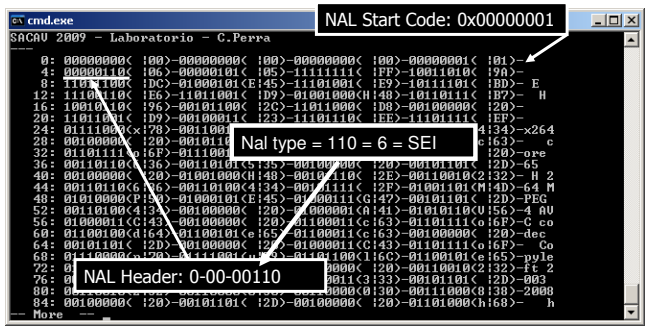


- E' possibile gestire più sequenze in un bitstream
- Una sequenza contiene più picture
- SPS e PPS sono numerate per identificare ogni sequenza e picture
- PPS ha un identificatore dell'SPS al quale fare riferimento
- Ogni VCL-NAL ha un identificatore del PPS al quale fare riferimento
- Il PPS deve essere inviato prima della slice di dati che fa riferimento a tale PPS

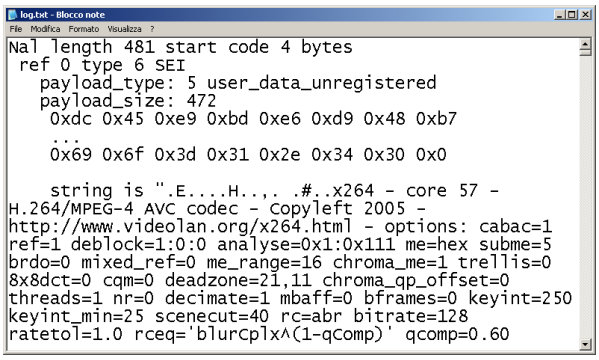
NAL units



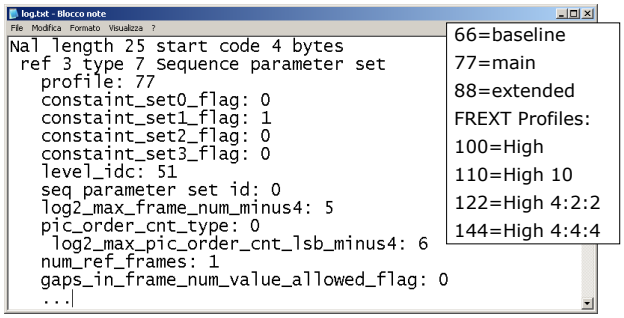
NAL unit stream (test.264)



Esempio: H.264 SEI



Esempio: H.264 SPS (1/3)



- Nota: le dimensioni effettive sono 27 byte.

Esempio: H.264 SPS (2/3)

log.txt - Blocco note

```
...
pic_width_in_mbs_minus1: 10 (176)
pic_height_in_map_minus1: 8
frame_mbs_only_flag: 1
  derived height: 144
direct_8x8_inference_flag: 1
frame_cropping_flag: 0
vui_parameters_present_flag: 1
 aspect_ratio_info_present_flag: 0
 overscan_info_present_flag: 0
 video_signal_info_present_flag: 0
 chroma_loc_info_present_flag: 0
 timing_info_present_flag: 1
  num_units_in_tick: 1
  time_scale: 50
  fixed_frame_scale: 1
...
```


Esempio: H.264 SPS (3/3)

log.txt - Blocco note

```
...
nal_hrd_parameters_present_flag: 0
vcl_hrd_parameters_present_flag: 0
pic_struct_present_flag: 0
motion_vectors_over_pic_boundaries_flag: 1
max_bytes_per_pic_denom: 0
max_bits_per_mb_denom: 0
log2_max_mv_length_horizontal: 11
log2_max_mv_length_vertical: 11
num_reorder_frames: 0
 max_dec_frame_buffering: 1
```


Esempio: H.264 PPS

log.txt - Blocco note

```
Nal length 8 start code 4 bytes
 ref 3 type 8 Picture parameter set
   pic_parameter_set_id: 0
   seq_parameter_set_id: 0
   entropy_coding_mode_flag: 1
   pic_order_present_flag: 0
   num_slice_groups_minus1: 0
   num_ref_idx_l0_active_minus1: 0
   num_ref_idx_l1_active_minus1: 0
   weighted_pred_flag: 0
   weighted_bipred_idc: 0
   pic_init_qp_minus26: 0
   pic_init_qs_minus26: 0
   chroma_qp_index_offset: 0
   deblocking_filter_control_present_flag: 1
   constrained_intra_pred_flag: 0
   redundant_pic_cnt_present_flag: 0
```

CAVLC

Il filtro di deblocking è abilitato

IDR e non-IDR

log.txt - Blocco note

```
Nal length 2176 start code 4 bytes
 ref 3 type 5 Coded slice of an IDR picture
   first_mb_in_slice: 0
   slice_type: 7 (I)
   pic_parameter_set_id: 0
   frame_num: 0 (9 bits)
   idr_pic_id: 0
   pic_order_cnt_lsb: 0
Nal length 34 start code 4 bytes
 ref 2 type 1 Coded slice of non-IDR picture
   first_mb_in_slice: 0
   slice_type: 5 (P)
   pic_parameter_set_id: 0
   frame_num: 1 (9 bits)
   pic_order_cnt_lsb: 2
Nal is new picture
```


non-IDR

log.txt - Blocco note

```
Nal length 181 start code 4 bytes
 ref 2 type 1 Coded slice of non-IDR picture
   first_mb_in_slice: 0
   slice_type: 5 (P)
   pic_parameter_set_id: 0
   frame_num: 2 (9 bits)
   pic_order_cnt_lsb: 4
Nal is new picture
Nal length 917 start code 4 bytes
 ref 2 type 1 Coded slice of non-IDR picture
   first_mb_in_slice: 0
   slice_type: 5 (P)
   pic_parameter_set_id: 0
   frame_num: 3 (9 bits)
   pic_order_cnt_lsb: 6
Nal is new picture
```


non-IDR

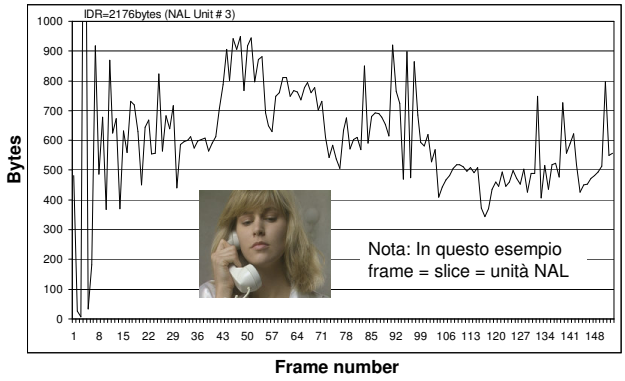
log.txt - Blocco note

```
...ultime due NAL unit
Nal length 797 start code 4 bytes
 ref 2 type 1 Coded slice of non-IDR picture
   first_mb_in_slice: 0
   slice_type: 5 (P)
   pic_parameter_set_id: 0
   frame_num: 147 (9 bits)
   pic_order_cnt_lsb: 294
Nal is new picture
Nal length 550 start code 4 bytes
 ref 2 type 1 Coded slice of non-IDR picture
   first_mb_in_slice: 0
   slice_type: 5 (P)
   pic_parameter_set_id: 0
   frame_num: 148 (9 bits)
   pic_order_cnt_lsb: 296
Nal is new picture
```

Riassumendo

- 1 Nal Unit SEI
- 1 Nal Unit SPS
- 1 Nal Unit PPS
- 1 Nal Unit IDR (frame I = 1 slice)
- 149 Nal Unit non-IDR (frame P = 1 slice)

Dimensioni delle NAL Unit



MPEG-4 AVC/AAC

- Wide adoption in IP-based video services over consumer broadband connections (IPTV)
- Several competing specifications for the transport of MPEG-4 over IP networks.
- Some IPTV deployments have utilized MPEG-2 Transport Stream (TS) for the carriage of MPEG-4 data (even though this approach was not designed to make use of MPEG-4-specific encoding structures)
- TS-based transport of MPEG-4:
  - TS over UDP/IP
  - TS over RTP/UDP/IP ("native RTP" transport of MPEG-4, the direct carriage of MPEG-4 encoded data in RTP packets)

MPEG-4 AVC

- AVC yields good video quality at bit rates currently achievable by ADSL and wireless connections
- Has received attention from standards bodies and industry focused on broadband and wireless video services

Standards bodies adopting AVC

AVC is currently adopted by the following standards bodies:

- ITU and MPEG put together the H.264/MPEG-4 part 10 specification.
- DVD Forum and the Blue-ray Disk Association selected AVC as mandatory for their respective next-generation high definition DVD formats.
- DVB adopted AVC for use in both standard and high definition digital television, as well as in wireless transmission to handheld devices.
- 3GPP selected AVC as the primary codec for mobile video in its Release 6 specification.
- ISMA defines profiles and specifications for streaming AVC over IP networks.
- AVC is under consideration for adoption by 3GPP2 and ATSC (adopted in September 2008).

AVC Transport

- The transport of AVC content is not uniform in these specifications.
- To aid in providing efficient and error resilient transport, the AVC specification defines a Network Abstraction Layer (NAL) that encapsulates the output of the encoder.
- NAL Units consist of video slices - independently-decodable groups of macro blocks with positioning, quantization and other data
- NAL Units form the basic fragments of video that are transmitted to clients.

AAC

- The Advanced Audio Codec was standardized by MPEG
- It is a high quality audio codec that significantly out-performs the well-known MP3 format
- It is currently used in Apple's iTunes software, as well as XM satellite radio, and is adopted by 3GPP, 3GPP2, the ISMA and DVB.

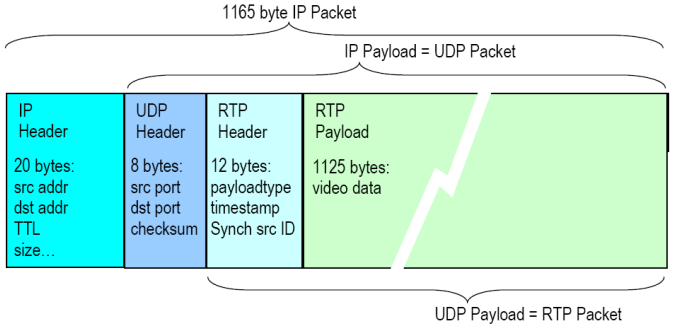
Internet Protocol

- The internet protocol (IP) is a packet-based-network transport protocol upon which the internet is built.
- IP packets are encapsulated in lower, hardware-level protocols for delivery over various networks (Ethernet, etc), and they encapsulate higher transport- and application-level protocols for streaming and other applications.

Internet Protocol

- IP packets consist of a header and a payload. The header contains addressing and control information that allows a packet to be routed through packet-switching networks.
- The payload contains the data that is to be transmitted. In the case of streaming over IP networks, multiple protocols, such as RTP and UDP (described below), may be carried in the IP payload, each with its own header and payload that recursively carries another protocol packet.

Example



MPEG-2 Transport Stream

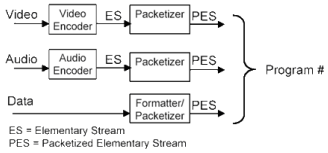
- Composed of 188 byte TS Packets, each with a 4 byte header
- Some TS packets contain an optional Adaptation Field whose size depends on flags set in the packet header and which may contain timing information, pad bytes, and other data.

MPEG-2 Transport Stream

- TS packet payloads may contain program information (PI) as well as Packetized Elementary Streams (PES), typically video and audio streams. PES packets are broken into 184 byte chunks to fit into the TS packet payload.
- They have a variable-length byte header which must coincide with the start of a TS packet payload. It is thus necessary to pad a TS packet that carries the last chunk of a PES packet when there is insufficient PES data to fill it.

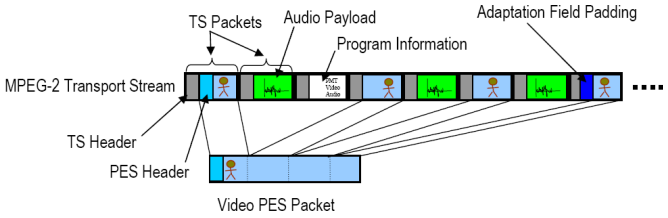
MPEG-2 Transport Stream

- A Transport Stream contains multiplexed data, carrying TS packets with payloads from multiple PES packets – again, typically audio and video – as well as associated program information.
- Because PES packet headers, as well as Adaptation Fields, contain timing information, no other signaling is necessary to synchronize multiple streams for playback.



MPEG-2 Transport Stream

- An MPEG-2 Transport Stream, multiplexing video, audio and program information.



Packetized Elementary Stream

- PES is a specification defined by the MPEG communication protocol (MPEG-2 standard) that allows an Elementary stream to be divided into packets. The elementary stream is packetized by encapsulating sequential data bytes from the elementary stream inside PES packet headers.
- A typical method of transmitting elementary stream data from a video or audio encoder is to first create PES packets from the elementary stream data and then to encapsulate these PES packets inside transport stream (TS) packets or program stream. The TS packets can then be multiplexed and transmitted using broadcasting techniques, such as those used in an ATSC and DVB.

Program stream

- **PS** or **MPEG-PS** is a container format for multiplexing digital audio, video and more. The PS format is specified in MPEG-1 Systems and MPEG-2 Part 1, Systems (ISO/IEC standard 13818-1). Program streams are created by combining one or more (PES), which have a common time base, into a single stream. It is designed for reasonably reliable media such as disks, in contrast to transport strean which is for data transmission in which loss of data is likely. Program streams have variable size records and minimal use of start codes which would make over the air reception difficult, but has less overhead.

Program stream

- Program streams are used on DVD video discs and HD DVD video discs. The file extensions are VOB and EVO respectively. Blu-ray Discs use a transport stream (TS) format with an additional 4 byte time code added to the beginning of each TS packet.

x264

- **MPEG-4 AVC / H.264 free encoder**
- **Stato:**
  - **Encoder features**
  - **CAVLC/CABAC**
  - **Multi-references**
  - **Intra: all macroblock types (16x16, 8x8, and 4x4 with all predictions)**
  - **Inter P: all partitions (from 16x16 down to 4x4)**
  - **Inter B: partitions from 16x16 down to 8x8 (including skip/direct)**
  - **Ratecontrol: constant quantizer, single or multipass ABR, optional VBV**
  - **Scene cut detection**
  - **Adaptive B-frame placement**
  - **B-frames as references / arbitrary frame order**
  - **8x8 and 4x4 adaptive spatial transform**
  - **Lossless mode**
  - **Custom quantization matrices**
  - **Parallel encoding on multiple CPUs**
  - **Interlacing**
  - **http://www.videolan.org/developers/x264.html**

Installazione (MinGW)

- MinGW-5.1.4.exe
- A collection of freely available and freely distributable Windows specific header files and import libraries, augmenting the GNU Compiler Collection, (GCC), and its associated tools, (GNU binutils). MinGW provides a complete Open Source programming tool set which is suitable for the development of native Windows programs that do not depend on any 3rd-party C runtime DLLs.
- http://sourceforge.net/project/showfiles.php?group_id=2435

Installazione (MinGW)

- MSYS-1.0.10.exe
- A Minimal SYStem providing a POSIX compatible Bourne shell environment, with a small collection of UNIX command line tools. Primarily developed as a means to execute the configure scripts and Makefiles used to build Open Source software, but also useful as a general purpose command line interface to replace Windows cmd.exe.
- http://sourceforge.net/project/showfiles.php?group_id=2435

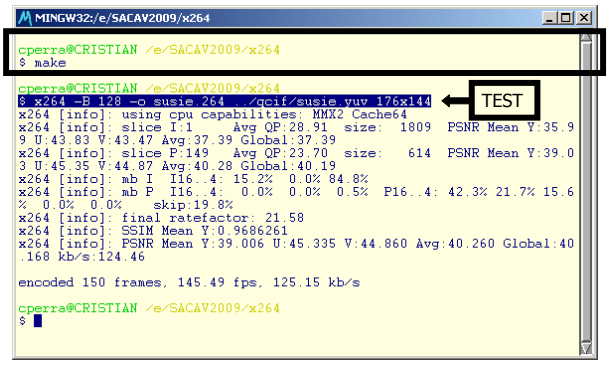
Installazione (MinGW)

- nasm-2.05rc2-win32.zip
- The famous Netwide Assembler
- http://sourceforge.net/projects/nasm

Preparazione del compilatore

1. Installare MinGW (C:\mingw)
2. Installare MSYS
3. Copiare nasm in c:\mingw\bin
4. Scaricare il codice x264:
   - p.e. dal tar giornaliero in ftp://ftp.videolan.org/pub/videolan/x264/snapshots/
5. Decomprimere il codice x264

Creazione dell'eseguibile



H.264/AVC Reference Software

- Software code, manual, documentation
  - http://iphome.hhi.de/suehring/tml/
  - Project: VisualStudio2005 / Linux
- Encoder
  - .yuv ➔ lencod ➔ .264
- Decoder
  - .264 ➔ ldecod ➔ .yuv
- RTP Packet content analysis
  - .264 ➔ rtpdump
- RTP Packet loss simulator
  - .264 ➔ rtp_loss ➔ .264

H.264/AVC Reference Software

- Configurazione encoder JM14.2 (encoder.cfg):
  - Files
  - Encoder Control
  - B Slices
  - Output Control, NALs
  - CABAC context initialization
  - Interlace Handling
  - Weighted Prediction
  - Picture based Multi-pass encoding
  - Deblocking filter parameters

H.264/AVC Reference Software

- (cont):
  - Error Resilience / Slices
  - Search Range Restriction / RD Optimization
  - Explicit Lambda Usage
  - Additional Stuff
  - Rate control
  - Fast Mode Decision
  - Rounding Offset control
  - Fast Motion Estimation Control Parameters
  - SEI Parameters
  - VUI Parameters

H.264/AVC Reference Software

- Configurazione decoder JM14.2 (decoder.cfg):
  - H.264/AVC coded bitstream
  - Output file, YUV/RGB
  - Ref sequence (for SNR)
  - Write 4:2:0 chroma components for monochrome streams
  - NAL mode (0=Annex B, 1: RTP packets)
  - SNR computation offset
  - Poc Scale (1 or 2)
  - Rate_Decoder
  - B_decoder
  - F_decoder
  - LeakyBucket Params
  - Err Concealment(0:Off,1:Frame Copy,2:Motion Copy)
  - Reference POC gap (2: IPP (Default), 4: IbP / IpP)
  - POC gap (2: IPP /IbP/IpP (Default), 4: IPP with frame skip = 1 etc.)
  - Silent decode
  - Enable Deblocking filter in intra only profiles (0=disable, 1=filter according to SPS parameters)

Esempio

Input YUV file	: ../susie.qcif
Output H.264 bitstream	: test.264
Output YUV file	: test_rec.yuv
YUV Format	: YUV 4:2:0
Frames to be encoded I-P/B	: 300/299
Freq. for encoded bitstream	: 15
PicInterlace / MbInterlace	: 0/0
Transform8x8Mode	: 1
ME Metric for Refinement Level 0	: SAD
ME Metric for Refinement Level 1	: Hadamard SAD
ME Metric for Refinement Level 2	: Hadamard SAD
Mode Decision Metric	: Hadamard SAD
Motion Estimation for components	: Y

Esempio

Image format	: 176x144 (176x144)
Error robustness	: Off
Search range	: 32
Total number of references	: 5
References for P slices	: 5
List0 references for B slices	: 5
List1 references for B slices	: 1
Sequence type	: I-B-P-B-P (QP: I 28, P 28, B 30)
Entropy coding method	: CABAC
Profile/Level IDC	: (100,40)
Motion Estimation Scheme	: Fast Full Search
Search range restrictions	: none
RD-optimized mode decision	: used
Data Partitioning Mode	: 1 partition
Output File Format	: H.264/AVC Annex B Byte Stream Format

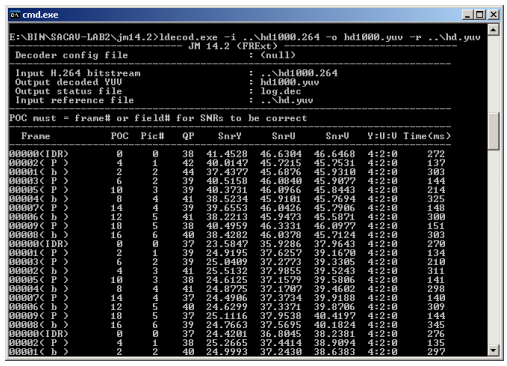
Esempio

Frame	Bit/pic	QP	SnrY	SnrU	SnrV
0000(NVB)	184				
0000(IDR)	16976	28	38.218	43.391	43.495
0002(P)	1080	28	37.550	43.326	43.491
0001(B)	256	30	37.844	43.362	43.454
0004(P)	2320	28	37.592	43.273	43.561
0003(B)	256	30	37.384	43.286	43.557
0006(P)	1016	28	37.510	43.295	43.546
0005(B)	96	30	37.555	43.257	43.545
0008(P)	1368	28	37.491	43.292	43.589

NVB, Nal NonVLC Bits

IDR, Intra frame

Esempio



Instantaneous Decoding Refresh

- (IDR) picture: A coded picture in which all slices are I or SI slices that causes the decoding process to mark all reference pictures as "unused for reference" immediately after decoding the IDR picture. After the decoding of an IDR picture all following coded pictures in decoding order can be decoded without inter prediction from any picture decoded prior to the IDR picture. The first picture of each coded video sequence is an IDR picture.

I slice

- A slice that is not an SI slice that is decoded using prediction only from decoded samples within the same slice.

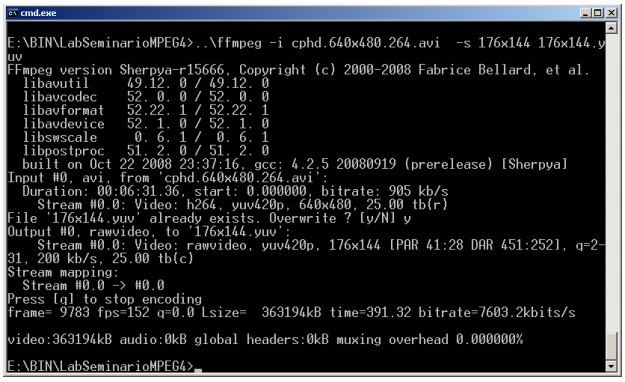
Download MPLAYER / FFMPEG

- http://sourceforge.net/project/showfiles.php?group_id=205275&package_id=248632

mplayer / mencoder / ffmpeg

- Playing yuv files:
  - mplayer -demuxer rawvideo -rawvideo w=176:h=144 susie.qcif
- Transcoding:
  - mencoder chd.avi -ovc x264 -vf scale=640:480 -o cphd.264 -nosound -x264encopts bitrate=1000
- Padding:
  - ffmpeg -vframes 50 -ss 00:03:56 -i chd.avi -padtop 220 -padbottom 220 -padleft 360 -padright 360 sacavHD.yuv
- YUV
  - ffmpeg -i chd.avi -s 176x144 176x144.yuv

Esempio (da avi a yuv)



Video Sequenza di Test

- **Filename**
  - 352x288.yuv
- **Type**
  - rawvideo, yuv420p, 352x288
  - 25.00 Hz
- **Length**
  - 9791 frames
- **Duration**
  - 00:06:31.64
- **Size**
  - 1.488.858.624 raw byte
  - 1,38GB)

MPEG2 e MPEG4v (ffmpeg)

MPEG2 Encoding (rate=200kbps)

-r 25 -s 352x288 -i 352x288.yuv -vcodec mpeg2video -b 200k -r 25 352x288x200kbps-mp2.mpg

MPEG2 Decoding

-i 352x288x200kbps-mp2.mpg -s 352x288 dec-mp2.yuv

MPEG4V Encoding (rate = 200kbps)

-r 25 -s 352x288 -i 352x288.yuv -vcodec mpeg4 -b 200k -r 25 352x288x200kbps-mp4.mpg

MPEG4V Decoding

-i 352x288x200kbps-mp4.mpg -s 352x288 dec-mp4.yuv

H.264 (x264)

- **H264 Encoding (rate = 200kbps)**
  - *x264 -B 200 -o 352x288.264 352x288.yuv 352x288*
  - (ffmpeg -r 25 -s 352x288 -i 352x288.yuv -vcodec libx264 -b 200k -r 25 352x288x200kbps-264.mpg)
- ***H264 Decoding***
  - *ffmpeg -i 352x288.264 dec-264.yuv*

H.264 (x264)

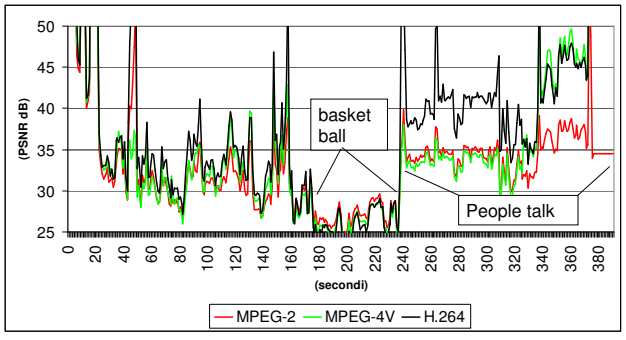
x264.exe -B 200 -o 352x288.264 352x288.yuv 352x288
using cpu capabilities: MMX2 Cache64
slice I:157 Avg QP:32.20 size: 5202 PSNR Mean Y:34.44 U:42.06 V:41.83 Avg:35.70 Global:30.87
slice P:9634 Avg QP:30.23 size: 856 PSNR Mean Y:37.51 U:44.84V:44.97 Avg:38.60 Global:32.86
mb I I16..4: 27.3% 0.0% 72.7%
mb P I16..4: 3.4% 0.0% 1.9% P16..4: 38.9% 8.0% 4.0% 0.0% 0.0% skip:43.8%
final ratefactor: 28.41
SSIM Mean Y:0.9240778
***PSNR Mean Y:37.462** U:44.797 V:44.921 Avg:38.555 Global:32.822 kb/s:185.18*

encoded 9791 frames, 38.91 fps, 185.33 kb/s

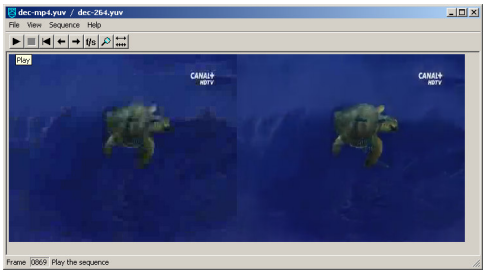
PSNR

MPEG-2	33.25dB
MPEG-4-Video	34.94dB (+5%)
H.264	37.43dB (+7%/+12%)

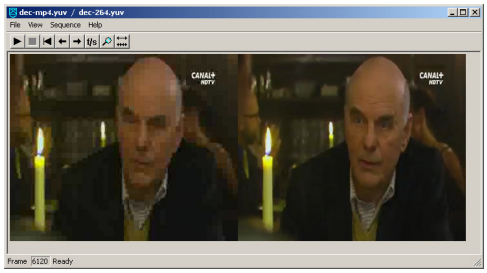
Confronto MPEG-2-4V-AVC



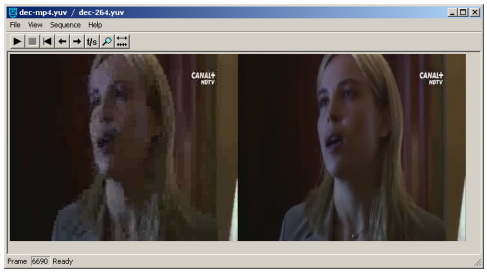
34.7sec



244.8sec



267.6sec



Acronimi

- ISO, International Organization for Standardization
- IEC, International Electrotechnical Commission
- DVB-S2, Digital Video Broadcasting - Satellite second generation, è uno standard del consorzio europeo DVB Project per la televisione digitale satellitare.
- DVB, Digital Video Broadcasting (DVB), insieme di standard aperti ed accettati a livello internazionale, concepiti per lo sviluppo e la diffusione della televisione digitale, mantenuti dal consorzio europeo DVB Project, pubblicati da un JTC dell'ETSI, del CENELEC, e dell'EBU.
  - Questi standard possono essere scaricati gratuitamente dal sito ETSI previa registrazione libera.

Acronimi

- JTC, Joint Technical Committee, Comitato Tecnico Congiunto
- ETSI, European Telecommunications Standards Institute , Istituto Europeo per gli Standard di Telecomunicazioni
- CENELEC, European Committee for Electrotechnical Standardization, Comitato Europeo per la Standardizzazione Elettrotecnica
- EBU, European Broadcasting Union, e dell'Unione Europea per la Radiodiffusione

Acronimi

- DVD, Digital Versatile Disc (Disco Versatile Digitale, originariamente Digital Video Disc, Disco Video Digitale)
- ES, Elementary Streams
- BIFS, Binary Format for Scenes
- OD, Object Descriptors